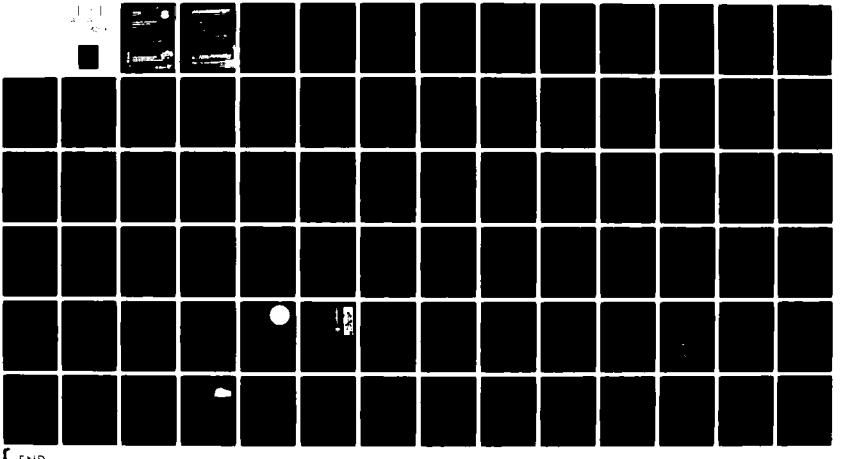
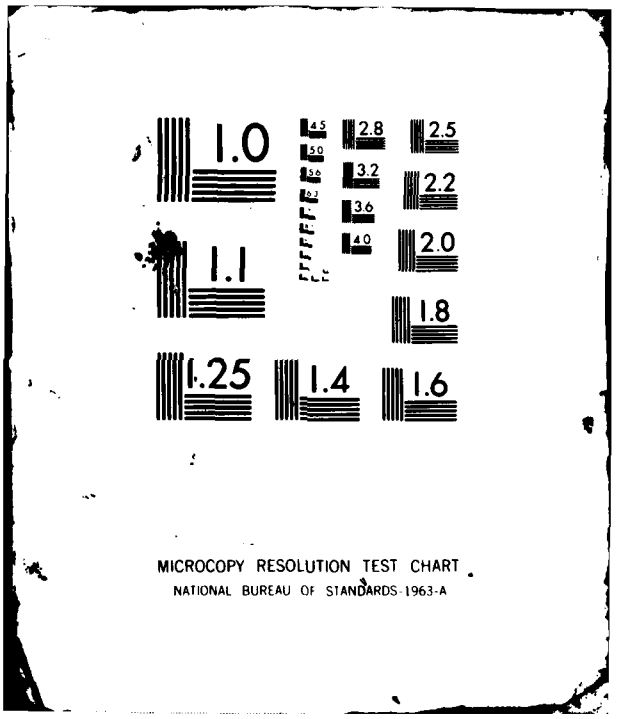
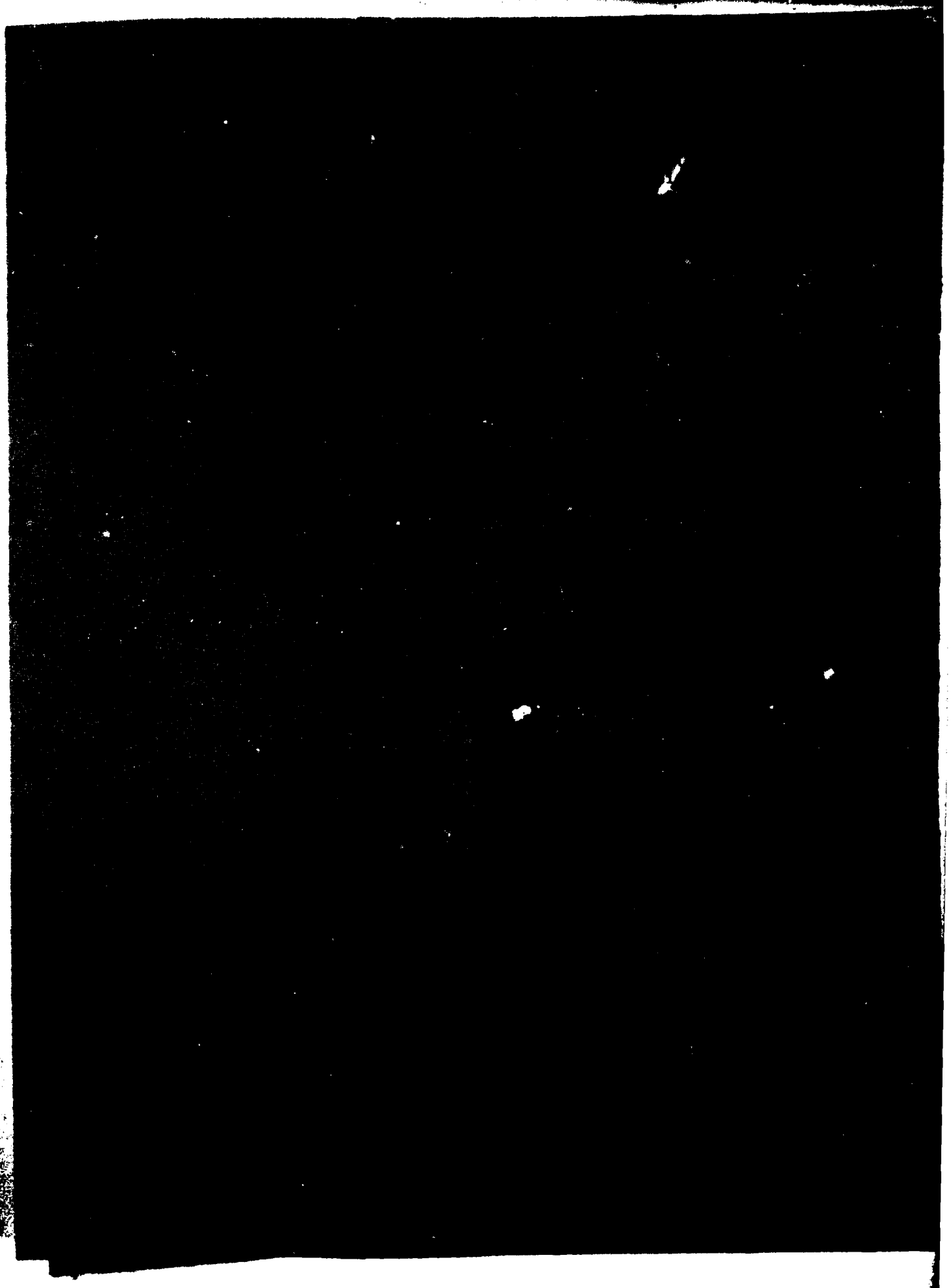
AD-A111 329 SRI INTERNATIONAL MENLO PARK CA F/G 9/2
AUTOMATIC HANDWRITING VERIFICATION (AHV).(U)
NOV 81 J S OSTREM, H D CRANE F30602-79-C-0255
UNCLASSIFIED RADC-TR-81-328 NL



END
DATE FILMED
3 82
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-81-328	2. GOVT ACCESSION NO.: AD-A111 329
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): AUTOMATIC HANDWRITING VERIFICATION (AHV)	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report Oct 79 - Mar 81
	6. PERFORMING ORG. REPORT NUMBER: SRI Project 8895
7. AUTHOR(s): John S. Ostrem, Hewitt D. Crane	8. CONTRACT OR GRANT NUMBER(s): F30602-79-C-0255
9. PERFORMING ORGANIZATION NAME AND ADDRESS: SRI International, 333 Ravenswood Avenue, Menlo Park CA 94025	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62710H, DNAR0217
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRAA), Griffiss AFB NY 13441	12. REPORT DATE: November 1981
	13. NUMBER OF PAGES: 84
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John V. Ferrante, 1Lt, USAF (IRAA)

This effort sponsored by the Defense Nuclear Agency

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Identify verification
Feature analysis
Correlation analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

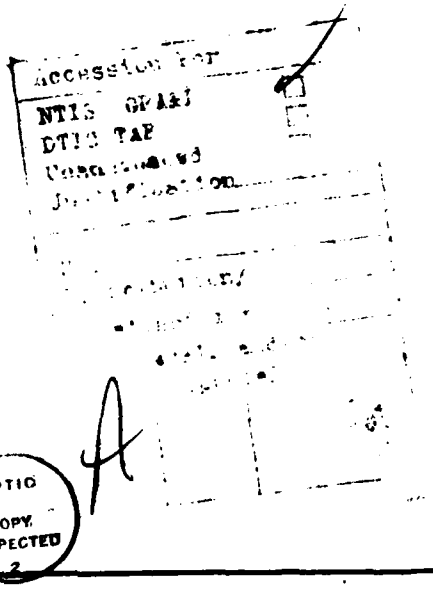
Over a four-month period, 5,220 signatures and 1,740 numeric "Signatures" were collected from 59 subjects writing with the SRI pen. Twelve trained "forgers" attempted 648 forgeries; they were given copies of the true-signer signatures as well as information about how the SRI system works and what measures would be used to perform the signature verification. The forgers were also allowed to practice their forgeries and to view video tapes of the true signers writing their signatures. Data were

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

analyzed to determine Type I/Type II error curves and average access time. For typical conditions, the True vs. Forgeries Type I/Type II equal-error rate was on the order of one percent, and the average access time, including the time to write the signature, was 8.5 seconds. These results are based on a features algorithm for signature verification using the same set of features for all subjects. It is shown that significant improvement may be achieved with the features technique if the feature sets are individualized for each user. Even greater improvement can be achieved with a correlation method of analysis, although this requires an increase in memory storage costs and computer processing time.



CONTENTS

APPENDICES

I INTRODUCTION

This report describes a one-year research program to evaluate the effectiveness of automatic signature verification based on three-axis signature dynamics.

There were two major aspects of the research effort:

- Data collection.
- Performance analysis to estimate the access time and Type I/Type II error curves for the signature verification system.

Over a four-month period, 5,220 signatures and 1,740 numeric sequences were collected from 59 subjects. These data were collected both with the subjects sitting at a table and standing at a counter. Twelve trained forgers attempted 648 forgeries. The forgers were given copies of the true signers' signatures, instructed in how the signature verification system works and what it measures, allowed to watch video tapes of the true signers writing their signatures, and allowed to practice as much as they desired over a three-week period. These data and the data collection protocol are described in detail in Section II.

Signature verification algorithms and associated data base analysis techniques are discussed in Section III. The primary focus is on the features and "rubbery" correlation algorithms for signature verification, and on a discriminant analysis approach to subject identification based on a handwritten sequence of numerals.

A detailed summary of the performance analysis results is given in Section IV. Estimates for the average access time and Type I/Type II error curves are presented for the features and rubbery correlation signature-verification techniques for a variety of operating conditions. The results of the subject identification trials based on a handwritten numeral sequence and a discussion of the user acceptability of the system are also given in Section IV.

A brief summary of the major results of the study and recommendations for future work are given in Section V.

Because of the proprietary nature of some of the software programs used in the research, copies of these programs and associated documentation will be delivered to RADC under separate cover. A magnetic tape containing all the data collected in a format compatible with RADC's PDP 11/70 operating under RSX 11-M will similarly be provided. Documentation for the magnetic tape, including a test program for reading data from the tape, is given in Appendix A of this report.

II DESCRIPTION OF THE DATA BASE AND DATA COLLECTION PROCEDURES

A. Summary of Data Collected

The collection of true-signer data took place over a four-month period from the beginning of June to the end of September 1980. A list of the 59 subjects who participated in the data base and the number of signatures for each is given in Table 1. Subjects are identified by their initials. A total of 5,220 true signatures was collected. These data will be delivered to RADC in the form of a magnetic tape in a format compatible with the PDP 11/70 computer (see Appendix A).

Table 1

NUMBER OF SIGNATURES AND NUMERAL STRINGS COLLECTED FROM EACH SUBJECT IN THE TRUE-SIGNER DATA BASE

Subject	Number of Signatures	Number of Numeral Strings	Total (Signatures + Numerals)
AAF	108	36	144
AEP	84	28	112
AEW	96	32	128
ASI	102	34	136
BEP	108	36	144
BJG	114	38	152
CAU	78	26	104
CBW	84	28	112
CEP	90	30	120
CMS	90	30	120
DEP	102	34	136
DRB	90	30	120
DWV	66	22	88
ELF	78	26	104
EMW	96	32	128
FET	96	32	128
FJM	72	24	96
FLL	102	34	136
GAN	96	32	128
GEG	96	32	128

Table 1 (Concluded)

Subject	Number of Signatures	Number of Numeral Strings	Total (Signatures + Numerals)
GEW	90	30	120
HEP	78	26	104
HFS	84	28	112
JCZ	66	22	88
JEE	90	30	120
JEM	96	32	128
JEP	102	34	136
JJS	114	38	152
JLP	102	34	136
JNH	84	28	112
JRL	90	30	120
KCN	84	28	112
KES	108	36	144
LAL	84	28	112
LEL	120	40	160
MAB	96	32	128
MAN	66	22	88
MER	42	14	56
MFA	84	28	112
MRC	108	36	144
OEK	102	34	136
PER	30	10	40
PES	66	22	88
PJP	120	40	160
PLH	78	26	104
RAB	102	34	136
RTK	54	18	72
RWH	102	34	136
RWR	90	30	120
SAW	108	36	144
SDJ	84	28	112
SEA	78	26	104
SEC	102	34	136
SEM	54	18	72
SRW	90	30	120
TDK	66	22	88
TPP	60	20	80
TSS	114	38	152
VKR	84	28	112
Total	5,220	1,740	6,960

The data base subjects were chosen at random from a large group of volunteers at SRI. The only constraints imposed on subject selection were to obtain an approximately equal number of women and men, about 10 percent left-handers (based upon estimates of the percentage of left-handers in the general population), and a range of heights, weights, and ages. Referring to Table 1, the left-handers in the data base are CMS, FET, PER, PES, RWH, and SEC. Thirty of the 59 subjects were women.

In addition to the signature data, 1,740 handwritten samples of the numeric sequence 12345 were obtained from the 59 subjects during the same data collection period. Although not specified in the original work statement, SRI, at the request of RADC, agreed to collect this numeral data for the purpose of determining how well the 59 subjects could be identified* from handwritten samples of the same set of characters. The numberic data collected are also summarized in Table 1. The total number of responses obtained (signatures and numerals) was 6,960.

In addition to the signature and numeric data, 648 forgery attempts were obtained from 12 trained forgers. A summary of the attempted forgery data is given in Table 2. The forgery data will also be delivered to RADC on magnetic tape (see Appendix A). A detailed discussion of what information was made available to the forgers and how they were trained is given in the next section (II-B).

The total amount of data collected, including signatures, numerals, and attempted forgeries, is on the order of 25 million bytes (25 megabytes or 200 megabits).

Finally, as a separate item, each subject in the true-signer data base was videotaped in the process of signing three signatures. As discussed in II-B, these tapes were used in the forger training to provide the kind of dynamic information that can be obtained by observing the true signer write his signature.

---

*"Verification" and "identification" have different goals. In verification a person makes a claim as to his identity and the system attempts to verify this claim by comparing his handwritten signature against the computer-stored reference or template of that person's known signature. In identification the person does not make an a priori claim as to his identity; rather, the system attempts to determine his identity by comparins his handwritten sample against the set of templates for all persons in the data base to find the closest match.

Table 2

SUMMARY OF FORGERY ATTEMPTS

Forger	True Signer	Number of Attempts
AEP	RWR	18
AEP	VKR	18
BEH	TPP	18
DEC	ASI	18
DEC	JNH	18
DEC	SAW	18
DEC	SDJ	18
GEM	ELF	18
GEM	JEE	18
JER	AEW	18
JER	GAN	18
JFL	DRB	18
JFL	MRC	18
JFL	RAB	18
PED	GEG	18
PEM	EMW	18
PEM	MAB	18
PEM	MFA	18
PEM	SEA	18
RWH	CMS	18
RWH	FET	18
RWH	PES	18
RWH	SEC	18
RWH	JEM	18
RWH	BJG	18
VEW	FJM	18
VEW	FLL	18
VEW	JRL	18
VEW	LEL	18
JSO	AAF	18
JSO	AEP	18
JSO	CEP	18
JSO	DEP	18
JSO	LAL	18
TPP	GEW	18
TPP	OEK	18
	Total	648

B. Data Collection Protocol

1. Data Collection Area

The data were collected in a partially enclosed area containing a table and a counter (a podium-like stand). For the reasons discussed below, at each data collection session the subject wrote signatures both while sitting down at the table and while standing at the counter. The operator (a research assistant) sat in front of a computer terminal immediately adjacent to the partially enclosed area. Although the area was partially enclosed, the subjects were not totally isolated from view nor acoustically shielded from the normal computer noise. In essence, the data collection environment was essentially what might be expected for a personal identification system used for access control to a computer area.

2. True-Signer Data Base

Upon entering the data collection area, the subject was given a standard form on which to write his signatures and numerals for the session. This form, shown in Figure 1, was filled out ahead of time with the subject's name, the date, and other pertinent information so that the subject was free to concentrate on signing his signature and writing the sequences of numerals. The operator told the subject whether the data collection session was to begin at the table or the counter. To avoid biases, the order of collection alternated; that is, at one session the standing signatures would be collected first and the next time the sitting signatures would be first. If the table was first, the subject wrote three signatures and one set of numerals (12345) sitting at the table, and then wrote three more signatures and another numeric sequence while standing at the counter. When the counter was first, the process was reversed. Thus a data collection session consisted of six signatures and two numeric sequences. Three signatures under both sitting and standing conditions were required for each data collection session, because in the performance analysis we planned to simulate a personal identification system that allowed up to three tries at verification.

During the first session, the subject was given brief instructions. He was told that the system measures forces and dynamics so that any unusual pauses in writing are likely to cause the signatures to be rejected. The subject was instructed to use his or her standard signature. A subject who typically used one or more signature variants (e.g., a full middle name one time and only an initial the next) was requested to use the most common version of the signature. The subject was instructed to inform the research assistant of any obvious mistakes such as leaving out a middle name or initial, or other gross signature variants. There were very few such mistakes and those that occurred were excluded from the data base.

The signature and numeral data for each session were collected in a real-time on-line basis. That is, whenever a subject wrote a signature it was automatically digitized by the PDP 11/40 computer and written out on a large disk (67 megabytes), including a header record consisting of the subject's initials, the date and time, a response or index number, and various other

Subject _____

Date _____

Data Record _____

SIGNATURES	RESPONSE NUMBER

COMMENTS:

FIGURE 1 FORM FOR DATA COLLECTION

pertinent information. All data forms for all subjects and for all data collection sessions, as well as hard-copy records of all program transactions, were saved. In all, sufficient records were maintained so that whenever questions arose about the data it was possible to reconstruct exactly what happened during the session in question.

For a signature verification system operating in the "real world," the users must cooperate with the system or risk being denied access to a secure area, computer account, or the like. However, no such motivation exists for a data collection effort of the type described here. Thus there is always the danger that subjects will grow careless after the initial novelty of the system wears off, which can lead to unnaturally large variations in the way signatures are written and cause an artificially low estimate of system performance (compared to a real-world system in which users are continually motivated by the need for access). Hence to better simulate real-world operating conditions we offered prizes for the signatures that were most consistent over the data collection period. The intent here was to provide at least some motivation for the subjects to perform as they would in a real-world environment.

3. Forger Data Base

The basic procedure for collecting and storing attempted forgery data was essentially the same as that for the true-signer data base described in the preceding subsection. This is to be expected, because in the real world there is no *a priori* knowledge as to who is the true signer and who is the forger, so both must be treated the same (up to the point of verification). For consistency, prizes ($100, $50, and $25) were also offered for the "best" forgeries to provide motivation for the forgers to practice and do the best job possible.

Since the forgery data collection procedure was essentially the same as that for the true-signer data base described above, it remains only to discuss the training and preparation of the forgers.

One of the first problems was the selection of forgers. This was difficult because the SRI signature verification system is based on the dynamics of a signature (i.e., the forces and motions used to create a signature) rather than its final static image.* Thus the requirements for being a successful forger in the SRI system are quite different than those for a "classical" forgery, whose purpose is to duplicate the static image of a signature. For example, in our system tracing a true signature would be one of the worst strategies for forgery, because tracing usually results in dynamics very different from those of the true signer even though the final result may be essentially identical. Our approach was, therefore, to select motivated people who had good manual dexterity and the capability of understanding the basic concepts behind the verification system.

---

*See Appendices B and C for further details.

Rather than requiring the forgers to make a few attempts at all the different signatures in the true-signer data base, we decided that a more realistic simulation of how a real forger would operate would be to have each of our forgers concentrate on three or four signatures. They were given several samples of these signatures and were also given a description of how the signature verification system operates: that it measures signature dynamics, that timing and forces are generally important, and that some of the typical features on which the verification is based are the total time of the signature, average force in the three orthogonal directions and the respective energies, the number of pen-ups and pen-downs, and so on. Each forger was allowed 18 attempts to forge a particular signature. After the first nine attempts he was shown a video tape with a close-up view of the subject signing his signature. This was intended to simulate the condition in which a real forger surreptitiously observes a person writing his signature to learn as much as possible about the dynamics of the signature. Before the actual forgery attempts, the forgers were allowed to practice as much as they wanted within a three-week period. In essence, the forgers were provided with all the information that a dedicated real-world forger could be expected to obtain.

C. Assessment of Data Quality

When a signature, set of numerals, or any other response is written using the SRI pen, the result is a set of three analog signals that are a time record of the instantaneous three-axis force* on the pen tip during writing. An example is shown in Figure 2. The question of data quality then has two aspects:

- How well the three analog time series signals represent the important characteristics of a handwritten signature.
- How accurate the digitized (discrete) representation is of the three analog time series signals that are generated using the analog-to-digital converter and PDP 11/40 computer.

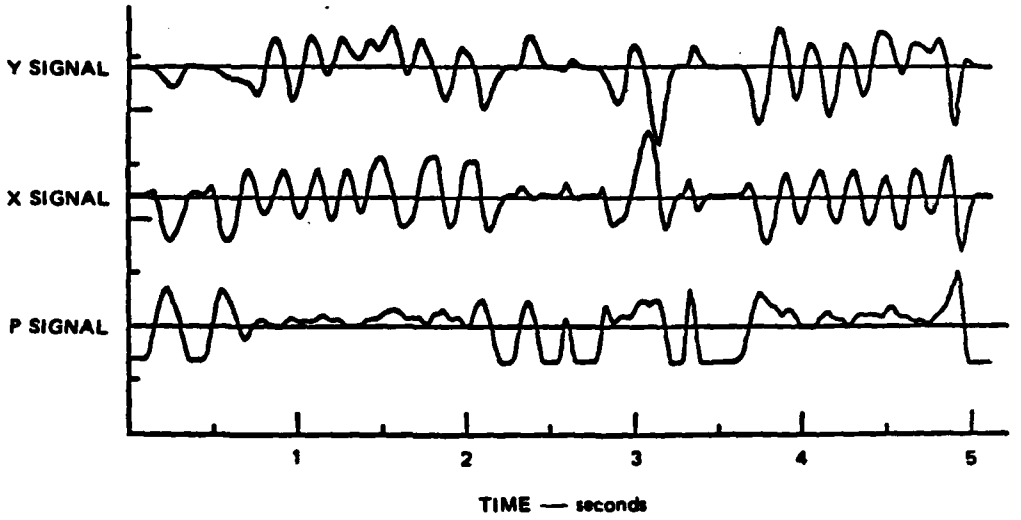
A discussion of the SRI pen as a device for transducing the motions used in handwriting into analog electrical signals representing the motions has already been published and hence will not be duplicated here. See Appendix B for details.

The data base was recorded and stored in digital (discrete) form. This approach was taken because it was more compatible with the subsequent processing and analysis, and because discrete data can be transported relatively simply between computers (e.g., in transferring the data base from SRI's PDP 11/40 to RADC's PDP 11/70). However, since we stored only the discrete representations of the P,X,Y analog signals, it was important to ensure

---

*I.e., force on the pen tip in three orthogonal directions. When the pen tip is vertical, the P-signal represents the downward force or pressure, and the X and Y force signals represent the left/right and far/near forces, respectively, in the plane of the writing surface.

(a) ORIGINAL SIGNATURE



(b) THE THREE-DIMENSIONAL SIGNALS GENERATED BY THE SRI PEN DURING THE WRITING OF THE ABOVE SIGNATURE

FIGURE 2 P, X, AND Y FORCE SIGNALS FOR A TYPICAL SIGNATURE

the accuracy of the discrete representations. The frequency response of the pen signals rolls off sharply above 25 Hz, because of filtering in the electronics, and it is reasonable to approximate the pen response as being frequency-bandlimited with a maximum frequency of about 25 Hz. The sampling theorem of communication theory states that for a bandlimited signal, sampling at least twice during the period of its highest frequency component is sufficient to completely characterize the signal in the sense that the original analog signal can be exactly reconstructed from the discrete samples. The minimum sampling rate for which this is possible is called the Nyquist rate, which corresponds to sampling exactly twice during each period of the highest frequency component. For the pen system the Nyquist rate = 1/2(25 Hz) = 0.02 s or 50 samples/s for each of the three analog signals. However, for safety we sampled at twice this rate, or 100 samples/s for each signal, for a total of 300 samples/s. This ensures that no loss of information occurs in the process of digitization and storage of the data in digital form.

During the data collection period, quality checks on the data were made at regular intervals. These checks consisted of printing out the data for selected signatures and displaying the P,X,Y signals in graphic form on a Tektronix display scope. As discussed earlier, a hard-copy record of all the data collection sessions was also maintained.

During the data collection period, only one software problem occurred. Under very unusual circumstances a very narrow spike was artificially introduced into the data. This problem, which was traced to a software error in the data collection program, was corrected at an early stage and affected only a very small amount of data. A computer program was written to search through all the data records to identify which were affected by this error. These responses were excluded from the analysis. There were occasions, also very infrequent, when the pen ran out of ink and had to be refilled, and on some such occasions a bad signature resulted. The number of bad responses from all sources is summarized in Table 3 for signatures and numerals.

The forgery data base collection began well after the true-signer data base, by which time these problems had been resolved, and hence all the forgery data was of good quality.

To summarize, of the total of 7,608 true signatures, numerals, and attempted forgeries, 47, or 0.6 percent, were deleted. The rest of the data was of high quality and was used in the subsequent analysis.

Table 3

BAD SIGNATURE AND NUMERIC RESPONSES

Subject	Response Numbers	
	Bad Signatures	Bad Numerals
AAF		
AEP		
AEW		
ASI	8, 11	
BEP		
BJG		4
CAU	34	
CBW		
CEP	33	
CMS	28, 60, 89	
DEP		
DRB	40	
DWV		
ELF	70, 76	
EMW	3	

Table 3 (Continued)

Subject	Response Numbers	
	Bad Signatures	Bad Numerals
FET	8	
FJM	49	
FLL	5	
GAN		
GEG		
GEW	55	
HEP	76	
HFS		
JCZ		
JEE		
JEM	1	
JEP	32, 36	
JJS	11	
MLP		
JNH	6	8
JRL		
KCN	12	
KES		
LAL		
LEL	19, 29	3
MAB		
MAN	16, 26	
MER	5	
MFA	62	
MRC	23	1
OEK		1
PER	3	
PES		
PJP	7, 26, 27, 39, 40	
PLH	2	
RAB	44	
RTK		
RWH		
RWR		
SAW		
SDJ		9
SEA	46	
SEC		
SEM		
SRW	2, 5	

Table 3 (Concluded)

Subject	Response Numbers	
	Bad Signatures	Bad Numerals
TDK		
TPP		
TSS		
VKR	12	

III DATA BASE ANALYSIS PROCEDURES

The purpose of the data base analysis is twofold:

- To optimize the performance of the signature verification system.
- To provide estimates of the performance of the optimized systems, including Type I/Type II error curves* and access time.

In this section we summarize the basic analysis procedures applied to the data base described in Section II. The results of the performance evaluation are reported in the next section (Section IV).

A. Features Analysis (for Signature and Forgery Data)

In this subsection we discuss the analysis procedures applicable to the features approach to signature verification. For background we begin with a description of how that approach actually works.

1. Features Approach to Signature Verification

The features approach to signature verification is summarized in Figure 3. When a person's identity is to be verified (e.g., to gain access to a secure area) the procedure is to identify himself to the system and write his signature. As shown in the figure, the pen transduces the forces and motions used in writing the signature into a set of three analog signals that are a time record of the instantaneous force on the tip of the pen in the three orthogonal directions. The P-signal is the downward force or pressure, and the X and Y signals are the left/right and far/near forces, respectively, in the plane of the paper. These analog signals are input to an analog-to-digital converter and digitized at the rate of 100 samples per second per channel. The digitized representations of the P,X,Y analog signals are then processed by a computer to extract a set of descriptors, called features, of the three signals. These features include various timing parameters such as the total time of the signature, the average force in each of the three directions (P, X, and Y) and the corresponding energies, the number of pen-ups and pen-downs, and so on. A complete listing of the features considered is given in III-A-2. The set of features (s_1, s_2, ... s_n) extracted from the discrete representations of the P, X, and Y signals form the feature vector when arranged in column order, as shown in Figure 3. The computer then calls up the computer-stored template or reference feature vector corresponding to the person whom the writer claims to be. The template vector is an average

---

*A Type I error occurs when a true signature is classified as a forgery (a false rejection). A Type II error occurs when a forgery is classified as a true signature (impostor accepted).

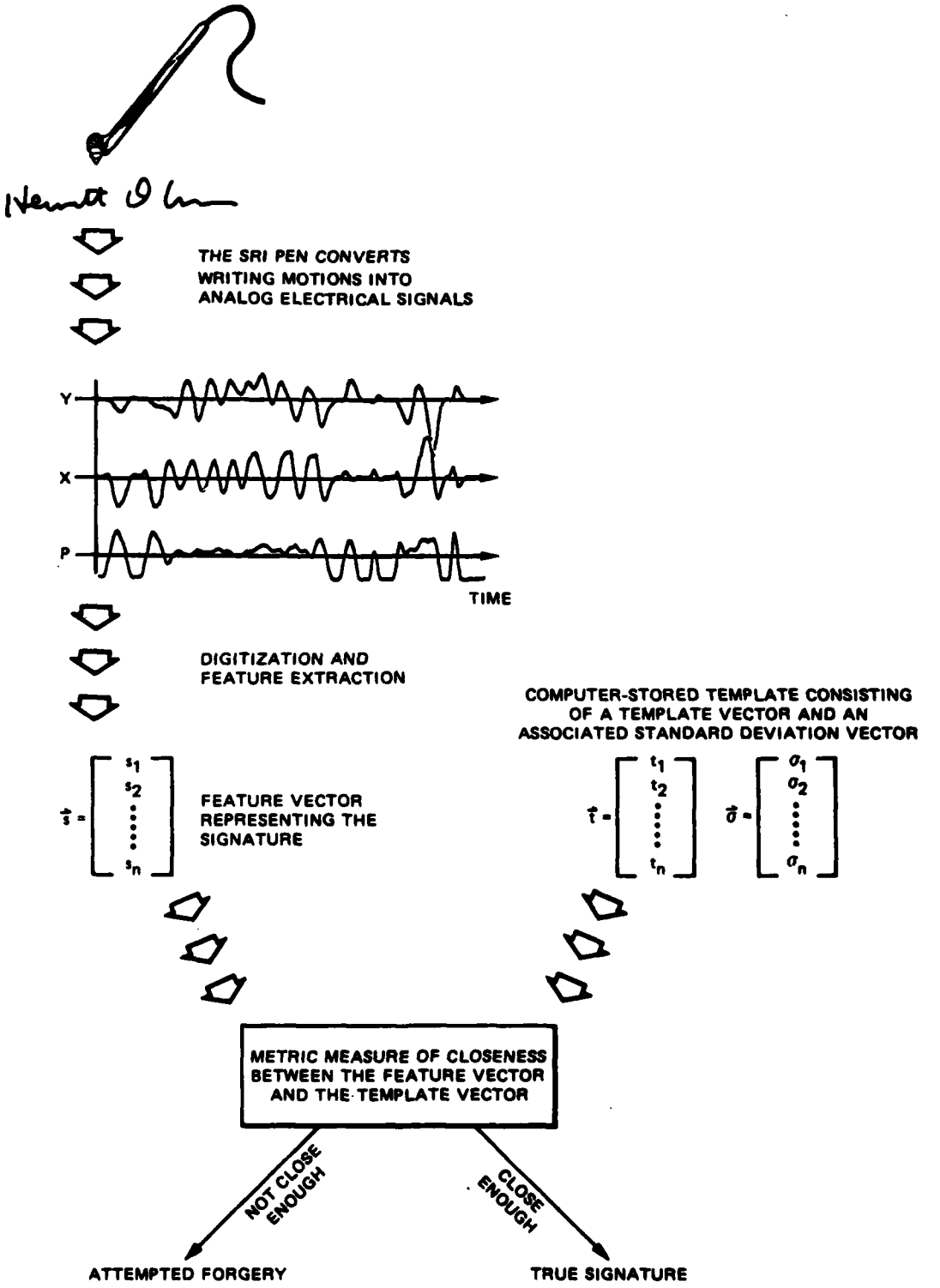


FIGURE 3 OVERALL VIEW OF THE FEATURES APPROACH TO SIGNATURE VERIFICATION

feature vector constructed from a set of known true signatures. Associated with the template vector is a vector of standard deviations for the features (see Figure 3), which provides a measure of how variable the true signer is from signature to signature for each feature. A measure of closeness between the feature vector and the template vector is then computed. If the feature vector corresponding to the signature in question is "close enough" to the template vector, the signature is judged to be a true signature and the person's identity is verified. If the feature vector is not close enough, the signature in question is judged an attempted forgery. In a practical signature verification system, the writer will often be allowed more than one chance to be verified; that is, if the first signature does not pass the above tests, he or she will be allowed to write one or two more signatures to be tested for verification.

A quantitative description of the computed "measure of closeness" between the feature vector and the template vector is given in Appendix C. In essence, the measure of closeness is a Euclidean distance metric, normalized or weighted by the template standard deviation vector. When the calculated distance metric is less than or equal to a pre-specified threshold, the signature is judged to be true; if above the threshold, it is judged to be an attempted forgery.

In the features technique, as we have seen, the forces and motions involved in creating a signature are finally represented as a feature vector. Clearly, if the features approach to signature verification is to be successful, the feature vector must contain as much information as possible that is useful for discriminating between true signatures and forgeries. The basic purpose for collecting a data base of true signatures and attempted forgeries is to provide data that can be analyzed to select a set of features (which constitute the feature vector) that provide maximum discriminating power between the true signatures and the attempted forgeries. The process of optimizing the features technique, then, consists of selecting the "best" set of features and an appropriate threshold for the distance metric measure of closeness (see Appendix C for details).

The problem of selecting a "best" set of features has two aspects, which we call feature extraction and feature selection. In general, there is no way to make an a priori determination of what the best features will be for a particular situation, so what must be done is to extract a relatively large number of features that are expected to be useful for discriminating between true signatures and attempted forgeries. This generally results in a great deal of redundancy. The objective of feature selection is to obtain a reduced set of features that contains essentially all the discriminating power of the original features set.* The features initially extracted for

*Under certain assumptions concerning the probability distributions of the feature set, it can be shown that the process of feature selection cannot reduce the Type I/Type II error rates. However, as a practical matter, an improvement in error-rate performance often results from feature selection. See R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (New York: Wiley, 1973), p. 66.

the signature verification problem are described in III-A-2. The feature selection process is discussed further in III-A-3.

2. Feature Extraction

Based on our knowledge of the characteristics of the P, X, and Y signals derived from the pen and our experience with previous true-signature and forgery data bases, a set of 44 features was selected as the starting point in the current data base analysis. These features are described in Figure 4.

FEATURE NUMBER			FEATURE
X	Y	P	
1	11	21	SCALED MEAN
2	12	22	STANDARD DEVIATION
3	13	23	MAXIMUM
4	14	24	MAXIMUM
5	15	25	AVERAGE ABSOLUTE
6	16	26	AVERAGE POSITIVE
7	17	27	NUMBER OF POSITIVE SAMPLES
8	18	28	AVERAGE NEGATIVE
9	19	29	NUMBER OF NEGATIVE SAMPLES
10	20	30	NUMBER OF 0-CROSSINGS
31	32	33	MAXIMUM MINUS SCALED MEAN
34	35	36	MAXIMUM MINUS MINIMUM
37	38	39	SCALED MEAN MINUS MINIMUM
FEATURE NUMBER			FEATURE
40			TOTAL TIME
41			NUMBER OF SEGMENTS -1
42			TIME UP
43			NUMBER OF SEGMENTS
44			TIME DOWN

FIGURE 4 THE 44 ORIGINAL FEATURES

3. Feature Selection

The objective of feature selection is the following: Given the relatively large set of 44 features described in the preceding subsection, find a subset that yields the best performance for signature verification. The process of feature selection yields two positive benefits:

- It reduces the Type I/Type II error rate.
- It improves computational efficiency by excluding or combining features that contain redundant information about the signature.

Before presenting the results of our analysis, we will give a brief description of general feature selection concepts.

In general, all feature selection techniques follow the same procedure. The starting point is a large set of features that the analyst believes to be useful for discriminating between samples (in our case, between true signatures and attempted forgeries). The discriminating power of each of the features, or subsets of features, is determined by performing statistical tests on a training set of data that is believed to adequately represent the population of interest. The subset of features that yields the best performance (by some criteria) and that contains the minimum number of features is the "best" feature set. Many procedures and algorithms for performing computerized feature selection have been devised. Some of these are based on univariate F-ratio evaluations,* Fisher's discriminate analysis,† information measures such as divergence,‡ and a host of ad hoc procedures. For the current project we tried a number of these techniques. Although some of them performed reasonably well, we were not entirely satisfied with the results. The standard feature selection techniques are all based on a number of assumptions about the underlying probability structure of the feature set. The exact assumptions differ somewhat from technique to technique, but in general it is assumed that the set of features is distributed as a multivariate Gaussian density, that the covariance matrices (see Appendix C for a definition of the covariance matrix) are equal, and the like. Our signature verification features do not appear to satisfy these conditions, and the result is that the feature selection techniques mentioned above do not operate in an optimum fashion; that is, there is no guarantee that the feature set obtained is the one that minimizes the Type I/Type II error rate. Because of the somewhat unsatisfactory performance of these classical feature selection

---

*The F-ratio technique for feature selection is described in many textbooks. For example, see W. J. Dixon and F. J. Massey, Introduction to Statistical Analysis, 3rd ed. (New York: McGraw-Hill, 1969), ch. 10; G. W. Snedecor and W. G. Cochran, Statistical Methods, 6th ed. (Iowa State University Press, 1967), ch. 14; and D. E. Bailey, Probability and Statistics (New York: Wiley, 1971), ch. 17 to 19.

†R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis (New York: Wiley, 1943).

‡S. Kullback, Information Theory and Statistics (New York: Wiley, 1959).

techniques, a new approach was devised that uses as its basic criterion the direct minimization of the Type I/Type II error rate.* This effort, which resulted in a much improved feature selection, can be summarized as follows: We began by extracting a subset of the total data base of signatures. Because we had an equal number of standing and sitting signatures, we typically used the standing signatures to select features (training the system) and the sitting signatures on which to make a final estimate of the error performance. To cross-validate the results the procedure was reversed, so that the sitting data were used as the training set and the standing data as the testing set. This process of using different sets of data to train and test the system provides a more realistic (conservative) estimate of the true error rates. (Using the same data for training and testing would yield unjustifiably optimistic results.) Starting with the full set of 44 features and (for example) the standing signature data as the training set, we first calculated the Type I/Type II error-rate curves† for all subsets of 43 features. We then examined the results to determine which of the 43-feature subsets yielded the best Type I/Type II error performance. Next we calculated the Type I/Type II error curves for all 42-feature subsets of the best 43-feature set, then for all 41-feature subsets of the best set of 42 features, and so on.‡ What typically occurs in this process is illustrated in Figure 5. As useless and/or redundant features are removed, the error rate decreases until it reaches a minimum. Once this minimum is reached, excluding more features results in reduced performance. The feature set that yields the minimum is selected as the best set.

The above procedure is an approximation to the more complete process of calculating the Type I/Type II error rates for all possible subsets of the 44 features, which is computationally prohibitive.§ Compared to the classical techniques for feature selection, this method has the following advantages:

- It requires no assumptions about the underlying probability distribution of the feature set.
- The calculations involved are relatively simple and intuitively reasonable.
- It selects a "best" feature set by choosing the subset that yields the least probability of error (subject to the qualification mentioned above that not all possible combinations of feature subsets are tested

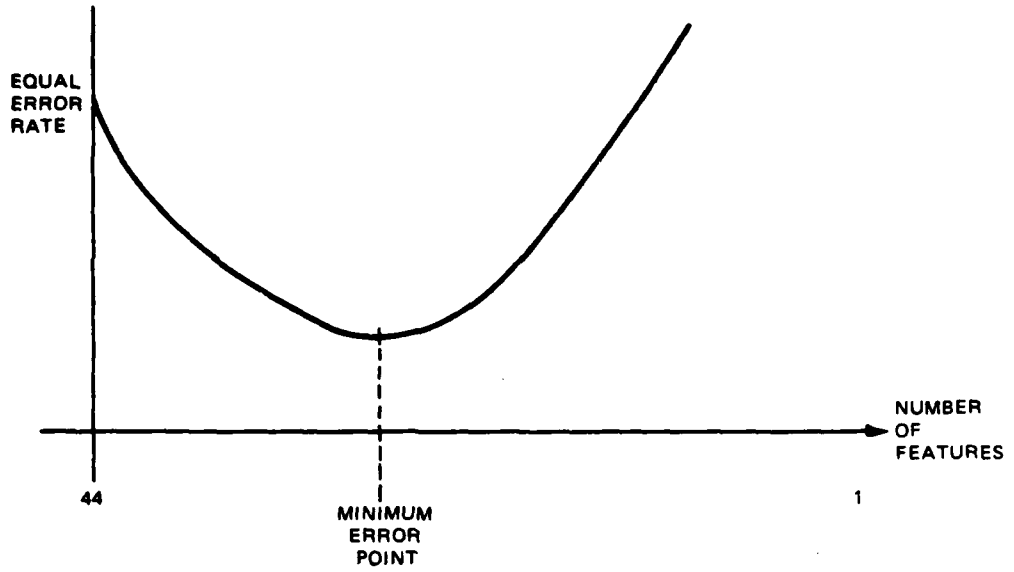
---

*Generally, the classical feature selection techniques cannot be related directly to Type I/Type II error rates except, as noted earlier, under a set of restrictive assumptions about the probability structure of the feature set (which are not satisfied by the signature verification features).

†The procedure for calculating these curves is described in IV-A.

‡This leave-one-out strategy can be rather time-consuming in itself. We were able to make the process more efficient by excluding more than one feature per iteration.

§A set of only 20 features would require more than one million Type I/Type II calculations.



*AT INTERSECTION OF THE TYPE I/TYPE II CURVES

FIGURE 5 TYPE I/TYPE II EQUAL-ERROR RATE* VERSUS NUMBER OF FEATURES

by our restricted search algorithm). This is not true of classical feature selection procedures in general, whose results can be said to minimize the probability of error only under a very restrictive set of assumptions, which experience has shown is not valid for the signature verification features.

- It takes into account correlations between features (implicitly). Redundant features (i.e., two features that are highly correlated) are excluded by the process of choosing the minimum point on the curve in Figure 5.

4. Type I/Type II Error-Curve Calculation Procedures

To calculate the Type I/Type II error curves, we developed an analysis procedure that simulates how a real-world signature verification system might operate. This program includes an enrollment phase in which templates are constructed from the first few (typically 10 or 12) signatures for each subject, and a verification phase in which subsequent true signatures and attempted forgeries are compared against the appropriate templates to determine the percentage of false rejections of true signers and of imposter

acceptances. The system allows up to three tries (signatures to be compared against the template) per verification trial. If the first signature fails to pass the verification criteria, a second signature is tested. If the second also fails, a third signature is considered. If all three signatures for a particular verification trial fail to pass the verification criteria, the subject is rejected as an impostor.

A template updating procedure was used to continually modify templates based on successful verification attempts. Each time a verification trial was successful on the first try (i.e., the first signature satisfied the verification criteria), the feature vector for that signature was added to the template vector with a weighting of 1/8. Thus if a subject's signature varied over time, the template would track the change. This template updating procedure was found to improve verification performance by reducing the percentage of true-signer rejections.

The basic criteria used to judge whether a particular test signature was a true signature or an attempted forgery was as follows: As in Figure 3, let $\vec{s}$ be the feature vector representing the test signature. The components of $\vec{s}$ are the values of the set of "best" features determined by the method described in the preceding subsection. Let $\vec{t}$ be the computer-stored template or reference vector and $\vec{\sigma}$ the associated standard deviation vector. The determinations of $\vec{t}$ and $\vec{\sigma}$ are based on an enrollment set of known true signatures (see Appendix C for explicit formulae for calculating $\vec{t}$ and $\vec{\sigma}$). Referring to Figure 3, the measure of closeness between the test signature and the template is the weighted Euclidean distance metric

$$d(\vec{s},\vec{t}) = \sqrt{\frac{1}{f}\sum_{i=1}^{f}\left(\frac{s_i - t_i}{\sigma_i}\right)^2} \quad ,$$

where f is the number of features, s_i is the value of the ith component or feature in the feature vector $\vec{s}$, t_i is the ith component of the template vector, and σ_i is the standard deviation of the ith feature as computed from a set of enrollment signatures. (See Appendix C for the reasons for selecting this Euclidean distance metric as the measure of closeness between the test signature and the template.)

The quantity $d(\vec{s},\vec{t})$ is a measure of closeness between the vectors $\vec{s}$ and $\vec{t}$. The smaller the calculated value of $d(\vec{s},\vec{t})$, the greater the similarity between $\vec{s}$ and $\vec{t}$, and therefore between the test signature represented by $\vec{s}$ and the computer-stored template $\vec{t}$ for the subject whose identity is being claimed by the person desiring to be verified.

The decision rule for deciding whether a particular test signature satisfies the verification criteria is:

- If $d(\vec{s},\vec{t}) \leq d^{thres}$, the signature is judged to be a true signature.
- If $d(\vec{s},\vec{t}) > d^{thres}$, the signature is judged to be an attempted forgery.

The quantity d^{thres} is a pre-assigned threshold value selected by using the Type I/Type II error curves to obtain the optimum trade-off between Type I and Type II errors for the particular application of interest. For example, for high-security applications, d^{thres} would likely be set to a relatively small value to minimize the impostor acceptance rate, while for banking applications in which the concern is usually to minimize user inconvenience (i.e., minimize the Type I error rate) a larger value for d^{thres} might be more suitable.

The procedure by which the Type I/Type II error curves were estimated from the data base is as follows: Let T^t represent the total number of verification trials in the true-signer data base, and let R^t represent the number of trials for which a true signer was falsely rejected. Note that the number of false rejections R^t is a function of the decision threshold while T^t is not. In general, R^t decreases as d^{thres} increases and increases as d^{thres} decreases. Recall that each verification trial allows up to three attempts, so that a false rejection occurs only when all three signatures fail to satisfy the verification criteria. The Type I error (false rejection rate for true signers) is estimated as

$$\text{Type I error} = \hat{E}_I = \frac{R^t}{T^t}$$

The $\hat{}$ symbol is used to indicate that $\hat{E}_I$ is an estimate of the error rate. When $\hat{E}_I$ is plotted as a function of the decision threshold, d^{thres}, the Type I error curve results. Similarly, the Type II error is estimated to be

$$\text{Type II error} = \hat{E}_{II} = \frac{R^f}{T^f}$$

where T^f is the total number of forger trials and R^f is the number of trials for which a forged signature passes the verification criteria (i.e., the number of imposter acceptances). R^f is also a function of the decision threshold; it increases with increasing d^{thres} and decreases with decreasing d^{thres}. A plot of $\hat{E}_{II}$ versus d^{thres} yields the Type II error curve. The justification for using the particular form of error-rate estimation given above is discussed in Appendix D. $\hat{E}_I$ and $\hat{E}_{II}$ are the maximum likelihood estimates (assuming independent trials) of the error rate for binomial distributed random variables.

$\hat{E}_I$ and $\hat{E}_{II}$ are estimates of the Type I and Type II errors, respectively, based on our data base of true signatures and attempted forgeries. The question then arises as to how confident we are that these estimates really correspond to the actual population error rates. In other words, our data base is only a sample drawn from a larger population of true signers and forgers, and we must ask how well we can estimate the true error rates for the larger population based on our particular sample. This leads to the concept of confidence limits, which is discussed in Appendix E. Basically if we say that

we have 95 percent confidence limits of ±1 percent for the Type I error rate, this means that we are 95 percent certain, given the estimated Type I error rate $\hat{E}_I$, that the true population error rate is within the range $\hat{E}_I$ ±1 percent. For example, if $\hat{E}_I$ = 2 percent then the true population error rate would be in the range 1 to 3 percent, with 95 percent confidence.

5. Individualized Feature Selection

In the preceding subsection we have discussed procedures for feature selection. These procedures can be used to determine a standard set of "best" features to be used for all subjects or to derive a set of best features for each subject individually. It is well known, both theoretically and from practical experience, that the use of individualized feature sets generally yields better signature verification performance* (lower Type I/Type II error rates), provided that enough training data is available to estimate the sets with reasonable statistical confidence. However, the use of individualized feature sets requires a more complex enrollment procedure, and it is not clear *a priori* that the improved performance is sufficient to justify its use for some practical applications of signature verification. In essence, the problem with individualized feature selection is that a large number of enrollment signatures is required from each subject to determine individualized feature sets with reasonable statistical confidence. If a standard feature set is used (i.e., if the same feature set is applied to all subjects) on the order of 10 to 12 signatures are adequate for enrolling a subject. This seems very practical and reasonable for a real-world signature verification system. Typically, to enroll in such a system a subject will sign five or six signatures on two different days. The requirements are quite different for individualized feature selection. Although the exact number of signatures needed cannot be determined without knowing the exact probability structure of the signature verification features (although they are definitely non-Gaussian), a standard rule of thumb in such instances is that the number of independent training (enrollment) samples be several times the number of features. Since we begin with 44 features, this implies that the number of enrollment signatures should be quite large, probably greater than 100, although it might be possible, with less confidence, to make do with 40 or so (perhaps even less if the set of features from which to choose is smaller). In any case, for a real-world application, this means that individualized feature selection may require a relatively long enrollment procedure. However, a compromise is also possible in which subject enrollment is based on a standard feature set, and, as more signatures are collected through subsequent verifications, the feature set is gradually and automatically individualized. But this approach has its own disadvantage, that of requiring the system to store, at least temporarily,

---

*This is intuitively reasonable. Since all subjects write differently, we would expect their signatures to be best characterized by somewhat different feature sets. For example, the total time that it takes to write the signature is a good feature for subjects who are consistent in the timing of their signatures but a bad feature for those who are very inconsistent in total writing time.

the large number of feature vectors needed for the process of individualized feature selection.

For the current project it was necessary to decide whether to focus the data base analysis on the features technique using a standard feature set or individualized feature sets for all subjects. Because of the magnitude of the data processing task (i.e., feature extraction, feature selection, and Type I/Type II error-curve calculations for many thousands of signatures) it was not possible to perform a complete and exhaustive analysis of both cases. For reasons discussed below, we decided to emphasize the standard set of features approach and process only a few problem subjects (i.e., the subjects with an abnormally high error rate) using individualized feature selection.

The data base collected for the project is, to the best of our knowledge, the first large-scale data base obtained using a three-axis signature verification system. Hence we decided it was most important to determine how well the basic signature verification system performed when using a standard set of features for all subjects. This approach also has the advantage that we can identify the small percentage of problem subjects from the standard feature set analysis and then apply the individualized feature selection process to determine what kind of improvement could be obtained for these problem subjects (discussed in IV-A-3). If we had started with the individualized feature set approach, there would have been no way to work backward to determine how well the system performed with a standard set of features.

B. Correlation Analysis

The features technique for signature verification has the advantage of simplicity and relatively low computational and template storage requirements. However, previous pilot studies indicate that the use of more sophisticated template-matching (i.e., verification) algorithms can result in substantially reduced Type I/Type II error rates. For high-security applications the potentially improved performance of a more sophisticated verification algorithm may outweigh the added complexity and computational requirements. In the following we describe SRI's "rubbery" correlation algorithm for signature verification.

In this algorithm, the P, X, and Y time-series force signals of a test signature are correlated mathematically with the appropriate P, X, and Y template signals. If the correlation is greater than or equal to a preassigned threshold (i.e., correlation value), the test signature is judged a true signature; if not, it is judged a forgery. However, a direct mathematical correlation generally yields rather poor performance (specifically, a high Type I error or true-signer rejection rate) because of the normal everyday variations in a person's signature. Even though the P, X, and Y signals for two signatures may seem highly correlated by a subjective visual comparison, there are often small time shifts within a particular test signature that cause significant misalignment between the prominent peaks and landmarks of the test signature P, X, and Y signals and the corresponding template signals. To compensate for the normal variations in a sequence of true signatures, we developed the technique of "rubbery" correlation, in which an automatic

two-dimensional search is used to find an optimal match (within appropriate search limits) between the test signature and the template, using various combinations of time-base translation and time-base warping (stretch and contraction) of the test signature P, X, and Y signals with respect to the template signals. These procedures can be applied independently to different parts of the signature--for instance, we can partition the test signature P, X, and Y signals into halves and correlate each half with the appropriate template P, X, and Y signals. It is also possible to use prominent landmarks (which are usually taken to be the pen-up intervals where the P signal, or downward pressure, is zero or close to zero) to partition a signature into smaller pieces on which to apply the time-warping algorithms.

The basic concept of rubbery correlation can be illustrated reasonably simply in one dimension (instead of three dimensions as is really the case when using the SRI pen): Let the template signal be represented as the vector $\vec{T}(t)$ whose components are the discrete sampled values of one of the analog signals obtained from a reference signature or template.

$$\vec{T}^{\dagger}(t) = [T_1, T_2, T_3, \ldots, T_n]$$

where † indicates the vector transpose and n is the total number of discrete samples. T_1 is the value of the template signal at time 1, T_2 is the value at time 2, and so on.

Let the test signal $\vec{V}(t)$ (obtained from a signature that is to be verified) be represented as

$$\vec{V}^{\dagger}(t) = [V_1, V_2, V_3, \ldots, V_n] \quad .$$

The standard Pearson correlation coefficient is defined as*

$$C[\vec{T}(t), \vec{V}(t)] = \frac{N \sum T_i V_i - \sum T_i \sum V_i}{\left[N \sum (T_i^2) - (\sum T_i)^2\right]^{\frac{1}{2}} \left[N \sum (V_i^2) - (\sum V_i)^2\right]^{\frac{1}{2}}}$$

The expression for correlation presented above is convenient for the purposes of explanation because its calculated values must lie between +1 and -1, where +1 and -1 are the maximum positive and negative correlations and 0 is no correlation. In practice there are more efficient ways to compute correlation if the -1 to +1 normalization is not required.

---

*T. W. Anderson, An Introduction to Multivariate Statistical Analysis (New York: Wiley, 1958), p. 49.

When $\vec{V}(t)$ is correlated against $\vec{T}(t)$, we judge $\vec{V}(t)$ to represent a true signature if the calculated correlation is larger than a preselected positive number, typically in the range 0.7 to 1.0. However, because true signers have some variability in their signatures, the correlation calculations must be made more flexible to allow a reasonable range of phase, amplitude, and time variations. This can be done by computing the correlation function

$$C[\vec{T}(t), \vec{V}(kt + t_0)]$$

for an allowed range of translation (i.e., for various t_0 in the above equation) and stretching or shrinking (i.e., for various values of the multiplicative constant k). The highest correlation over a specified range of discrete values of k and t_0 is thus obtained. If this correlation is larger than a specified value, the test signature is judged to be a true signature. Further flexibility is obtained by breaking the signature into pieces, either in fixed proportions such as halves or by using signal landmarks such as pen-ups, correlating each piece allowing for the k and t_0 variations described above, and combining them into a total correlation coefficient.

The procedure for calculating the Type I/Type II error curves for the rubbery correlation signature verification algorithm is essentially the same as described in III-A-4 for the features technique. The only difference is that the measure of closeness between a test signature and the template is now the rubbery correlation rather than the Euclidean distance metric.

C. Features Analysis (for Numeric Sequence Data)

The objective of collecting and analyzing handwritten numeric sequences was to determine how well subjects could be discriminated on the basis of handwritten samples of the same set of characters. As mentioned earlier, this is an identification problem rather than a verification problem because it is assumed that the subject makes no a priori claim as to his identity. In verification, the subject makes an a priori identity claim and the test sample is compared only against the computer-stored template (or reference) corresponding to the claimed identity. In identification, the subject writes a test sample that is compared against the templates of all the subjects to establish his identity.

Our analysis of the numeric sequences is based on the 44 features described in III-A-1. The set of 44 features was extracted from each of the 1,740 numeric sequences in the data base (see II-A for a description of that data base) using our PDP 11/40 and written to magnetic tape. A computer program was written to translate this tape into a format compatible with SRI's CDC 6400 computer. The feature data was then analyzed using the Statistical Package for the Social Sciences (SPSS) supported by the CDC 6400.

The SPSS was used because it is ideal for the kind of identification or classification problem posed by the numeric sequences. The specific program used for the current analysis, known as DISCRIMINANT, is based on standard

discriminant analysis procedures for classifying unknown test samples into one of many groups. Since this program is very well documented* and is available on most large-scale computers, we will not discuss it in detail here. The results of the identification analysis are given in IV-C.

*N. H. Nie *et al.*, SPSS, *2nd ed.* (*New York*: McGraw-Hill, 1975).
M. J. Norusis, "SPSS Statistical Algorithms (Release 8.0)," Computer Software for Data Analysis, Suite 3300, 444 N. Michigan Ave., Chicago, Illinois 60611.

IV PERFORMANCE EVALUATION

In Section III we described the signature verification algorithms and data base analysis procedures. In IV-A we summarize the results of the data base analysis, in terms of the Type I/Type II error curves and average access (or verification) time, for the features techniques using a standard feature set that is the same for all subjects. For typical conditions the equal-error rate* is on the order of one percent. In IV-A-3 and IV-B we show the improvement in performance that may be obtainable by using individualized feature selection and the rubbery correlation algorithm, respectively. In IV-C we present the results of a subject identification study based on a sequence of handwritten numerals, and in IV-D we discuss the human engineering aspects of the process (i.e., how the subjects felt about using the system).

A. Features Technique for Signature Verification

The procedure for selecting features and estimating the Type I/Type II error curves was discussed in III-A-3 and III-A-4, respectively. The set of 44 features (descriptors of the P,X,Y force signals generated by the SRI pen during the writing of a signature) used in the analysis was also described in III-A-3. In this subsection (IV-A-1) we begin by deriving the average time required for verification (i.e., the average access time). In IV-A-2 we present Type I/Type II error curves based upon a standard set† of "best" features derived from the original set of 44 features.

1. Access Time

The average signature length of the 58‡ data base subjects is 5.7 seconds. Added to this is a 1.5 second delay that is used to determine when the signature has been completed (i.e., no writing for 1.5 seconds indicates the signature is over) and a processing time of 0.5 seconds. The processing time varies with the length of the signature, and we have taken a worst-case estimate. The signature verification system allows up to three tries (signatures)

---

*As discussed later in more detail, the equal-error rate is the error rate at which the Type I/Type II error curves intersect (i.e., where Type I error = Type II error).

†By a standard set of features we mean a single set of features that is used for all subjects.

‡Subject PER was excluded from the data base analysis because other commitments prevented him from participating for the full length of the data collection period, and too few signatures of his were available to be analyzed.

per verification trial, but the analysis shows that on the average only 1.1 attempts were required. The average access time is thus estimated to be

$$\text{Average access time} = (5.7 + 1.5 + 0.5) \times 1.1$$
$$\approx 8.5 \text{ seconds}$$

2. Type I/Type II Error Curves

In this section we present Type I/Type II error curves beginning with the so-called "Trues vs. Trues" error curves. These curves are calculated based on the following: The known true signatures of a particular subject, say subject ABC, are compared against his own template. The percent rejection as a function of the decision threshold is the Type I error curve.* The Type II error is calculated by comparing the true signatures of all the other subjects against the ABC template. The percent accepted as a function of decision threshold is the Type II error curve. This procedure is then repeated for all subjects in the data base. Clearly the Trues vs. Trues error rate is a kind of confusion rate, comparable to the situation in which one subject claims the identity of another subject but attempts to use his own signature for verification. However, this is not a very realistic measure of the system's Type I/Type II error curves and is included here only because this type of error-rate calculation is very common in the literature. Following the presentation of Trues vs. Trues Type I/Type II error curves, we present the Trues vs. Attempted Forgeries Type I/Type II error curves. In this case the Type I error curves are calculated in the same way as the above, but the Type II error curves are computed using attempted forgery data.

a. Trues vs. Trues

The initial set of 44 features was described in III-A-2. To select a "best" subset of the 44 features we began by dividing the signature data into two sets, a training set and a testing set. Because we collected an equal number of sitting and standing signatures,† a natural division was made on this basis. To begin we used the sitting signature data as the training set on which feature selection was performed in order to determine a best subset of the 44 original features (i.e., the subset that yields the least error rate). Using the feature selection method described in III-A-3, the best subset consisted of Features 1, 2, 3, 6, 11, 12, 13, 14, 16, 20, 22, 25, 26, 27, 28, 29, 30, 32, 33, 38, 40, 41, 42, 43, and 44. These features are described in III-A-2. The standing data was then used to calculate the Type I/Type II error curves, the result of which is shown in Figure 6. To compare results we will use the point at which the Type I/Type II errors are equal (i.e., the

---

*See III-A-4 for more details. Recall that the Type I/Type II errors are calculated based upon allowing three tries (signatures) per verification trial.

†Signatures were obtained from subjects both sitting down at a table and standing at a counter. See Section II for details.

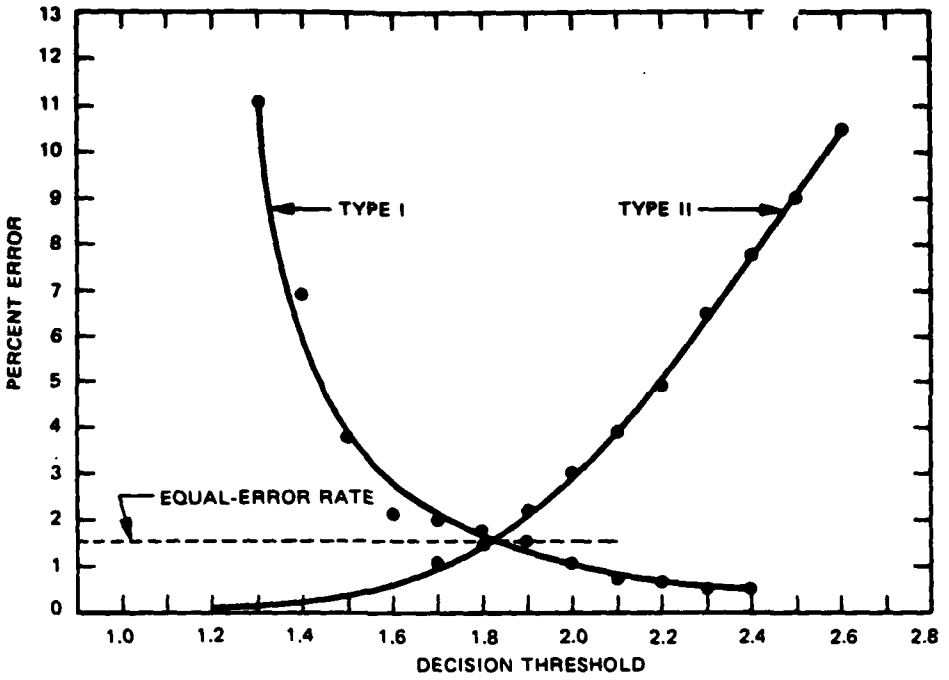


FIGURE 6 TYPE I/TYPE II ERROR CURVES FOR TRUES VERSUS TRUES, STANDING DATA

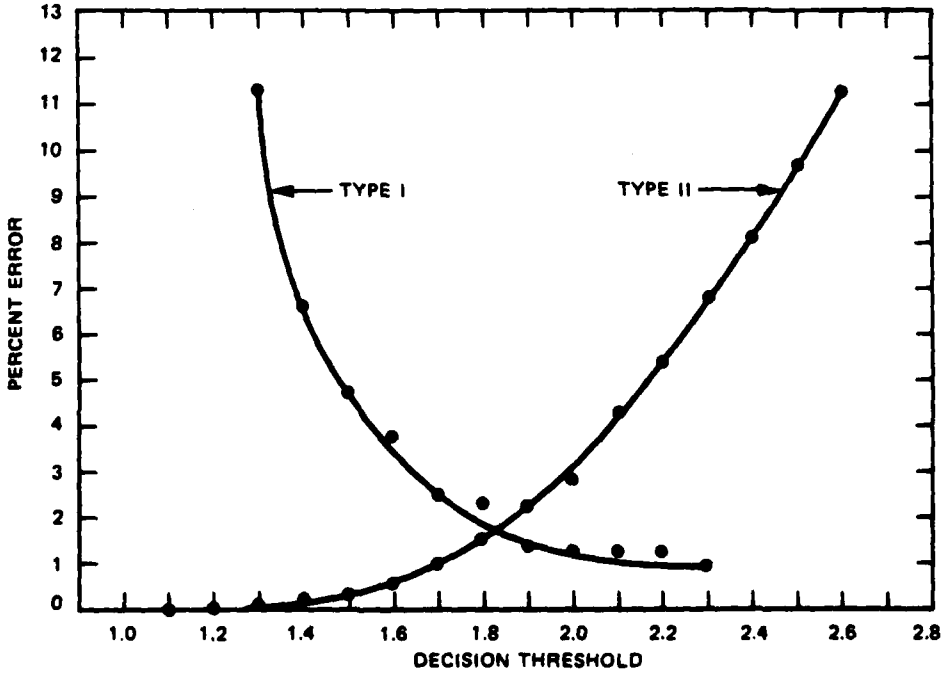


FIGURE 7 TYPE I/TYPE II ERROR CURVES FOR TRUES VERSUS TRUES, SITTING DATA

percent error where the curves intersect), which is called the equal-error rate. This equal-error rate, indicated by the horizontal dashed line in Figure 6, is about 1.5 percent for the standing data. To cross-validate the results we reversed the roles of the sitting and standing data. In this case the standing data were used to train (i.e., for feature selection) and the sitting data for error-rate calculation. The feature set selected using the standing data was the same as had been derived using the sitting data. The Type I/Type II error curves for the sitting data are shown in Figure 7. Comparison of Figures 6 and 7 show the error curves to be essentially identical, so the cross-validation yielded very consistent results, which gives us added confidence in the results. It may also be concluded that there is essentially no difference in performance whether the subject is sitting or standing when he writes.

b. Trues vs. Attempted Forgeries

For the Trues vs. Attempted Forgery* Type I/Type II error-curve calculations we decided to use the same set of best features that had been used for the Trues vs. Trues calculations. The reason for this is that the generality of the forgery data is uncertain because very little is known about the forger population. In any case, the use of the Trues vs. Trues feature set is a conservative approach, and there is no question of testing and training on the same data set.

The Type I/Type II error curves for the standing true-signature data versus the attempted forgery data are shown in Figure 8. The equal-error rate is approximately 2.25 percent, somewhat worse than the 1.5 percent equal-error rate of the Trues vs. Trues data. The Type I/Type II error curves for the sitting Trues vs. Attempted Forgeries is shown in Figure 9. The equal-error rate is almost 3 percent.

Data analysis showed that almost all the forgeries occured for the two or three true signers who were the most inconsistent in writing their signatures. A simple enrollment criterion based on the total variance of the template was subsequently tested. If the combined standard deviation was larger than some assigned threshold, the subject failed the enrollment criteria and was excluded. This resulted in the exclusion of three subjects out of 58 and yielded considerable improvement in signature verification performance.† Figures 10 and 11 show the Type 1/Type II error curves (for standing and sitting data, respectively) when this enrollment criterion is used. The

*In attempting to forge a signature each forger was allowed up to 18 tries, nine before viewing the video tapes and nine after. Because we found that there is only a slight difference in the error rates for the two conditions, the Type II error curves presented in this section are calculated using the combined set of forgery attempts.

†This behavior is typical of verification systems. Usually most of the errors are contributed by a very small percentage of system users.

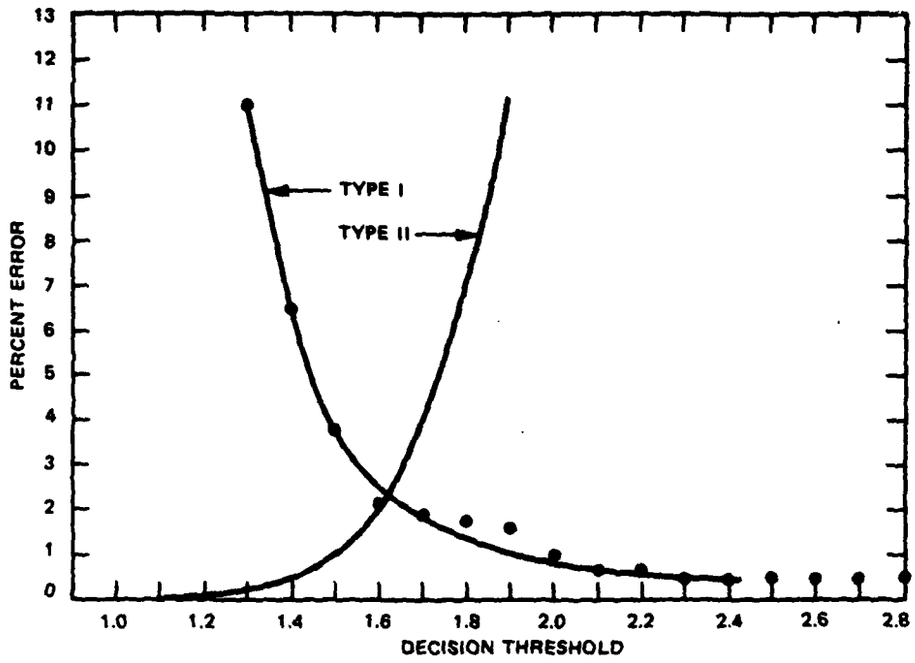


FIGURE 8 TYPE I/TYPE II ERROR CURVES FOR TRUES VERSUS ATTEMPTED FORGERIES, STANDING DATA, NO ENROLLMENT CRITERIA

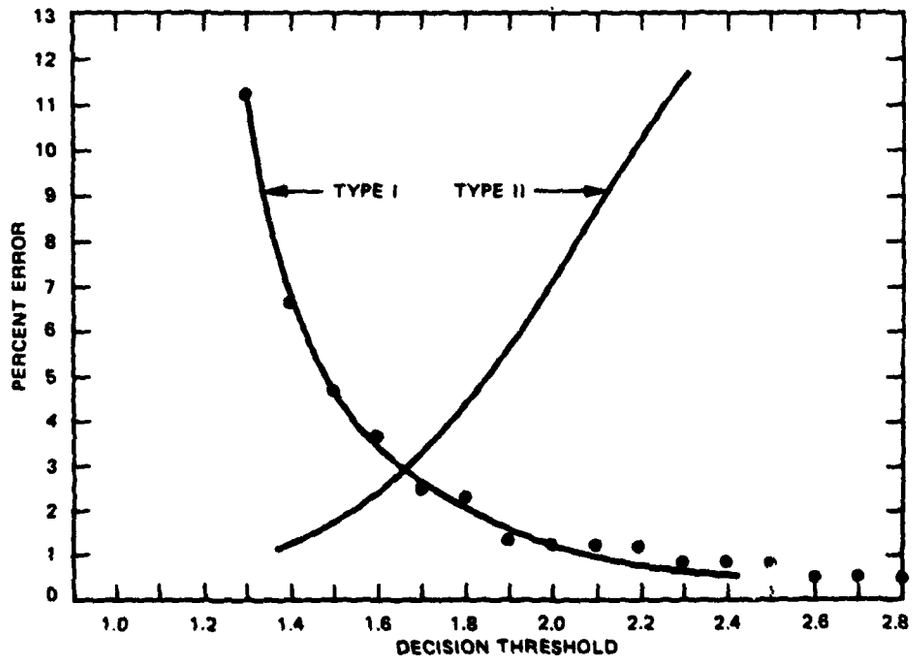


FIGURE 9 TYPE I/TYPE II ERROR CURVES FOR TRUES VERSUS ATTEMPTED FORGERIES, SITTING DATA, NO ENROLLMENT CRITERIA

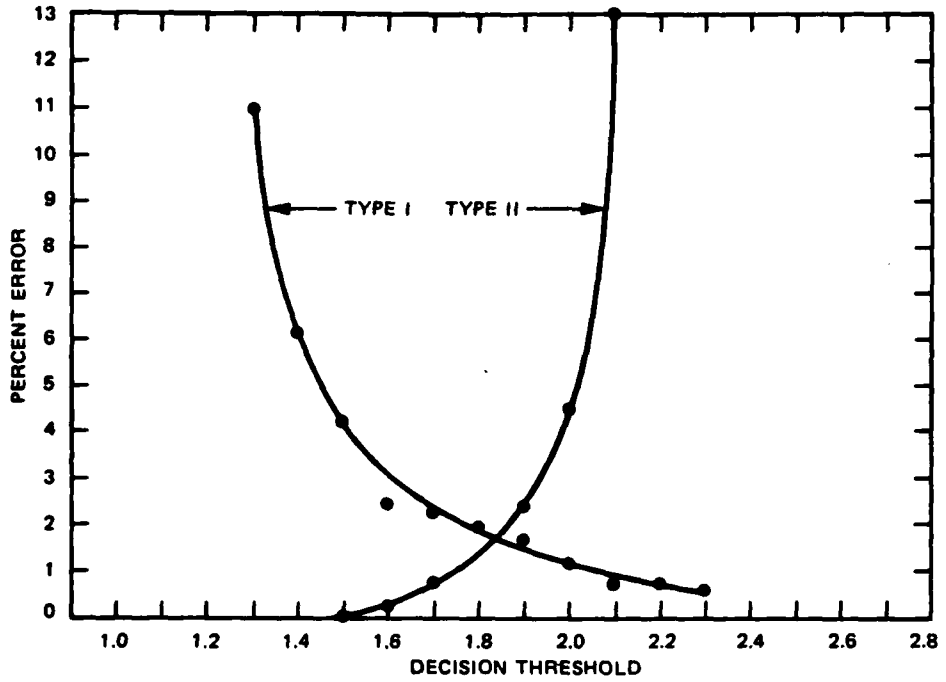


FIGURE 10 TYPE I/TYPE II ERROR CURVES FOR TRUES VERSUS ATTEMPTED FORGERIES, STANDING DATA, WITH ENROLLMENT CRITERIA

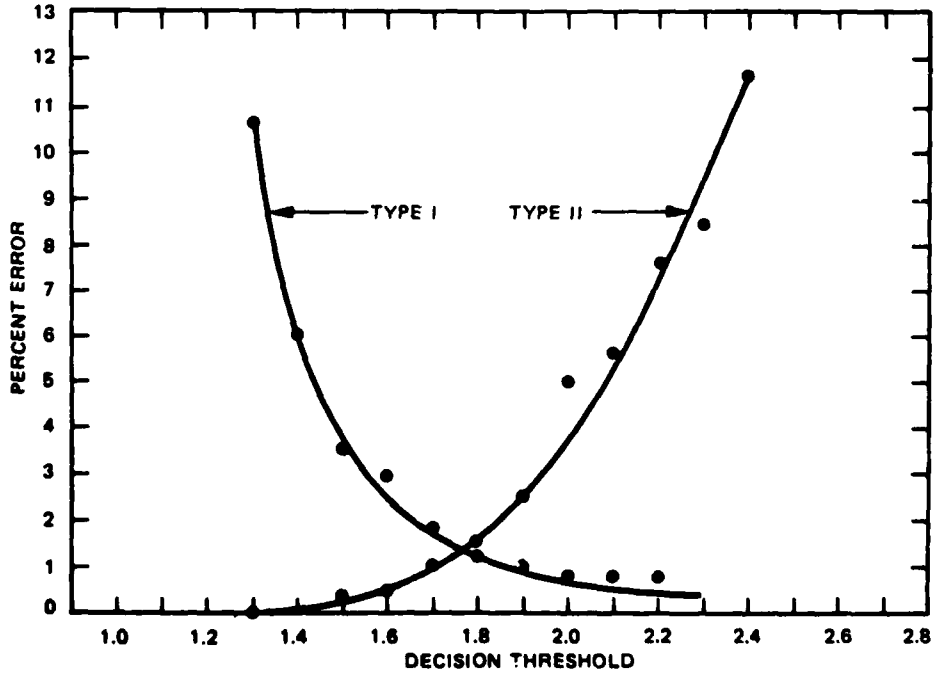


FIGURE 11 TYPE I/ TYPE II ERROR CURVES FOR TRUES VERSUS ATTEMPTED FORGERIES, SITTING DATA, WITH ENROLLMENT CRITERIA

equal-error rates are reduced to 1.75 percent for the Trues vs. Forgeries (standing) and to 1.25 percent for the Trues vs. Forgeries (sitting). By making the enrollment criteria even more stringent, to where six or seven of the most inconsistent subjects (out of 58) are excluded, the equal-error rates are on the order of 0.5 to 0.75 percent.

Tests were also made of the effect of allowing the forgers to view video tapes of the true signers writing their signatures. The error rate was slightly worse when the forgers were allowed to view the video tapes, which implies that the forger can learn something of the signature dynamics by closely observing the true signer. The total effect, however, was not particularly significant.

3. Performance Results for Individualized Feature Sets

As discussed earlier, improved performance can be expected when individualized feature sets are used. In this section we show, by example, the kind of improvement that can be expected. Of all the data base subjects, CMS was the worst in the sense of contributing the most to the Type I/Type II error rates. In Figure 12, the solid lines are the Type I/Type II error curves for subject CMS's true signatures vs. attempted forgeries using the standard feature set described in IV-A-2. The equal-error rate is over 6 percent.

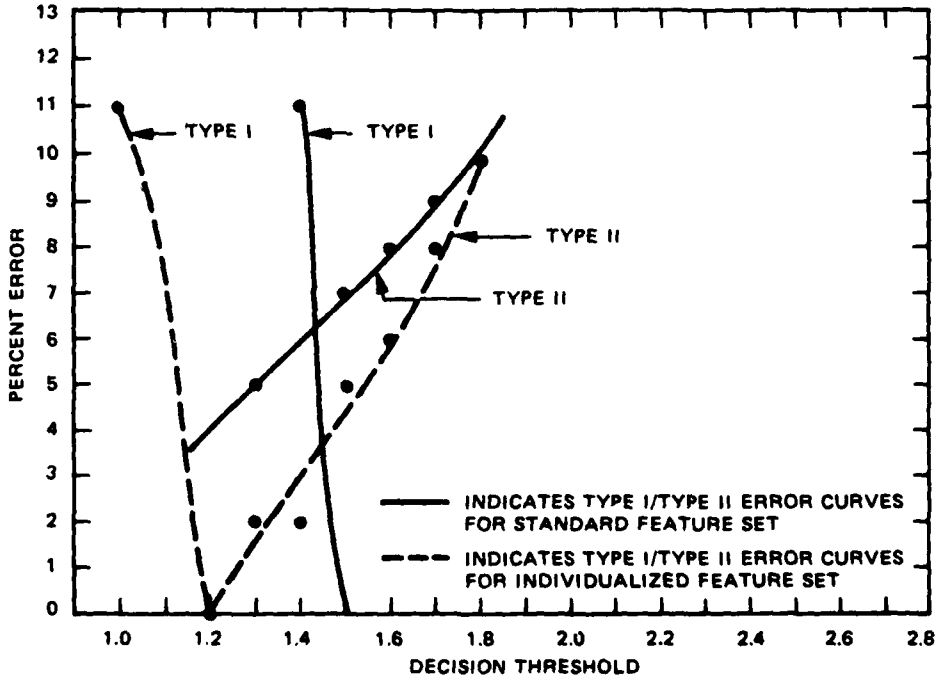


FIGURE 12 TYPE I/TYPE II ERROR CURVES FOR SUBJECT CMS FOR THE STANDARD FEATURE SET AND FOR AN INDIVIDUALIZED FEATURE SET

The individualized feature set for CMS, which was derived using the method described in III-A-3, consisted of Features 1, 2, 6, 11, 13, 16, 26, 27, 32, 38, 40, and 44.* The Type I/Type II error curves for subject CMS using this individualized feature set are given by the dashed curves in Figure 12. Note the substantial improvement compared to the Type I/Type II error curves for the standard feature set. In fact, for the individualized feature set there is no cross-over at all of the Type I/Type II error curves, and so the equal-error rate is zero. However, this is based on a small amount of data (one subject's true signatures and the associated attempted forgeries) and it would not be appropriate without extensive further testing to conclude that individualized feature selection would yield a Type I/Type II error rate of zero. However, based on this result and previous experience, we believe (but have not proven) that a conservative statement of the improvement which could be expected from individualized feature selection is that the equal error rate would be reduced by at least a factor of two (i.e., the equal-error rate would be on the order of 0.5 percent or better rather than the 1 percent as given in the preceding subsection).

B. Correlation Technique for Signature Verification

The rubbery correlation algorithm for signature verification was described in III-B. Because of time limitations and the fact that our PDP 11/40 computer was down with hardware problems for more than two months, we were unable to process the entire data base using the correlation algorithm. However, the main question is whether the rubbery correlation technique is more effective than the features technique for signature verification. To answer this question we processed true vs. attempted forgery data for those subjects for which the features technique yielded relatively poor performance. As discussed in IV-A-3, subject CMS contributed a high percentage (more than 6 percent) of the errors that occurred with the features technique. For the same set of data, subject CMS's Type I/Type II error curves for the rubbery correlation signature-verification algorithm are shown in Figure 13. These results may be compared with the Type I/Type II error curves (indicated by the solid lines in Figure 12) for the features technique. There is no overlap in the curves in Figure 13 and so the equal error rate is 0, a dramatic improvement.

Although we were not able to process enough data with the correlation algorithm to give a statistically confident estimate of the Type I/Type II error curves, our tests with some of the problem subjects, such as that for CMS described above, suggests very strongly that the correlation technique is substantially superior to the features technique for signature verification.

As noted earlier, the problem with individualized feature selection is the requirement for a long enrollment period with many signatures. However, this is not a problem for the correlation algorithm. The results for subject

*These features are described in III-A-2.

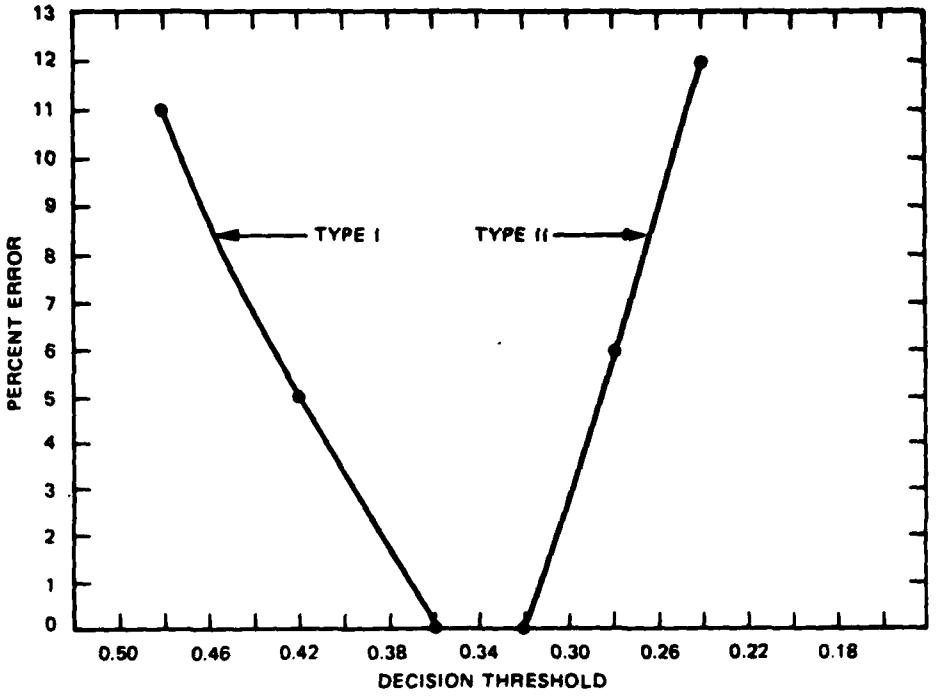


FIGURE 13 TYPE I/TYPE II ERROR CURVES FOR SUBJECT CMS (TRUES VERSUS ATTEMPTED FORGERIES) FOR THE "RUBBERY" CORRELATION SIGNATURE-VERIFICATION ALGORITHM

CMS described above were based on using only the first nine signatures for enrollment. The only disadvantage of the correlation technique, compared to the features technique, is somewhat increased processing time and increased computer storage requirements for the subject templates. For high-security applications, these disadvantages are probably not very important.

C. Features Technique for Subject Identification Based on a Handwritten Sequence of Five Numerals

In this section we present the results of the analysis of the handwritten numeric sequence data base using the SPSS program DISCRIMINANT.* The SPSS control file† used for the data analysis, which is shown in Figure 14, was set

*See *III-C* for the reasons that we chose to use the SPSS programs for the numeric sequence analysis, as well for references relating to program documentation and data analysis algorithms.

†The use of the SPSS control file and the many program options is described in detail in N. H. Nie, et al., SPSS, 2nd ed. (New York: McGraw-Hill, 1975).

```
RUN NAME        RADC SIGNATURE DATA, FEATURES SELECTED 11-MAY-81 AT 08:46:35
VARIABLE LIST   AUTHOR,SEQUENCE,POSITION,TRUEFORG,FEATUR01 TO FEATUR44
INPUT MEDIUM    DISK
INPUT FORMAT    FIXED(2F3.0,2F2.0,7F10.3        , /,10X,7F10.3, /,10X,7F10.3
                          , /,10X,7F10.3, /,10X,7F10.3, /,10X,7F10.3
                          , /,10X,2F10.3)
VALUE LABELS    AUTHOR  ( 1)AAF ( 2)AEP ( 3)AEW ( 4)ASI ( 5)BEP ( 6)BJG
                        ( 7)CAU ( 8)CBW ( 9)CEP (10)CMS (11)DEP (12)DRB
                        (13)DWV (14)ELF (15)EMW (16)FET (17)FJM (18)FLL
                        (19)GAN (20)GEG (21)GEW (22)HEP (23)HFS (24)JCZ
                        (25)JEE (26)JEM (27)JEP (28)JJS (29)JLP (30)JNH
                        (31)JRL (32)KCN (33)KES (34)LAL (35)LEL (36)MAB
                        (37)MAN (38)MER (39)MFA (40)MRC (41)OEK (42)PER
                        (43)PES (44)PJP (45)PLH (46)RAB (47)RTK (48)RWH
                        (49)RWR (50)SAW (51)SDJ (52)SEA (53)SEC (54)SEM
                        (55)SRW (56)TDK (57)TPP (58)TSS (59)VKR/
                POSITION (0)STAND (1)SIT/
                TRUEFORG (0)TRUE (1)FORGER/
N OF CASES      UNKNOWN
SEED            STANDARD
COMPUTE         WGTVAR=1
IF              (UNIFORM(1) LE 0.5)  WGTVAR=0
WEIGHT          WGTVAR
PRINT FORMATS   FEATUR01,FEATUR11,FEATUR21,FEATUR41,FEATUR44 (3)
LIST CASES      CASES=100/
                VARIABLES=AUTHOR,SEQUENCE,POSITION,TRUEFORG
                          ,FEATUR01,FEATUR11,FEATUR21,FEATUR41,FEATUR44/
READ INPUT DATA
DISCRIMINATE    GROUPS=AUTHOR(1,59)/
                VARIABLES=FEATUR01 TO FEATUR44/
                ANALYSIS=FEATUR01 TO FEATUR44/
                METHOD=DIRECT/
                PRIORS=EQUAL/
OPTIONS         5,6,10,11,12,20
STATISTICS      1,2,3,4,6
```

FIGURE 14 SPSS CONTROL FILE

up so that the DISCRIMINANT proram used approximately half (by random selection) of the 1,740 numeric sequences in the data base for training (i.e., to estimate the discriminant functions) and the other half for testing (error-rate calculations).

The basic result was that 90.4 percent of the numeric sequences in the testing data set were classified correctly; that is, 90.4 percent of the time a subject was identified correctly based upon a single handwritten numeric sequence. This recognition rate can be improved by allowing the subject to try again if his first handwritten sequence fails to identify him correctly.* The 95 percent confidence limits on the 90.4 percent recognition rate are ±2 percent.†

---

*For example, assuming independence, the recognition rate allowing two trials would be 99.1 percent.

†90.4 percent is an estimate of the true recognition rate for the population. The confidence limits simply state that we are 95 percent sure, given the estimate of 90.4 percent calculated from the data, that the true population recognition rate is between 88.4 and 92.4 percent.

Figure 15 presents a summary of the identification results on a subject-by-subject basis. The vertical column of initials is the actual author of the numeric sequence and the horizontal row of initials along the top gives the initials of the subject identified as the author (which may or may not correspond to the true author, depending on the success of the identification process). For example, the first subject on the vertical column of subject initials is AAF. Looking across that row we see that 17 numeric sequences of AAF were tested and all correctly identified as having been written by AAF; all 17 responses are listed under the column headed AAF. Similarly, there were 12 total numeric sequences tested for subject AEP and all 12 of them were identified correctly as having been written by AEP. Reading across the row for AEW, we see that there was a total of 15 numeric sequences tested. Of these, 14 were identified correctly and one was incorrectly identified as

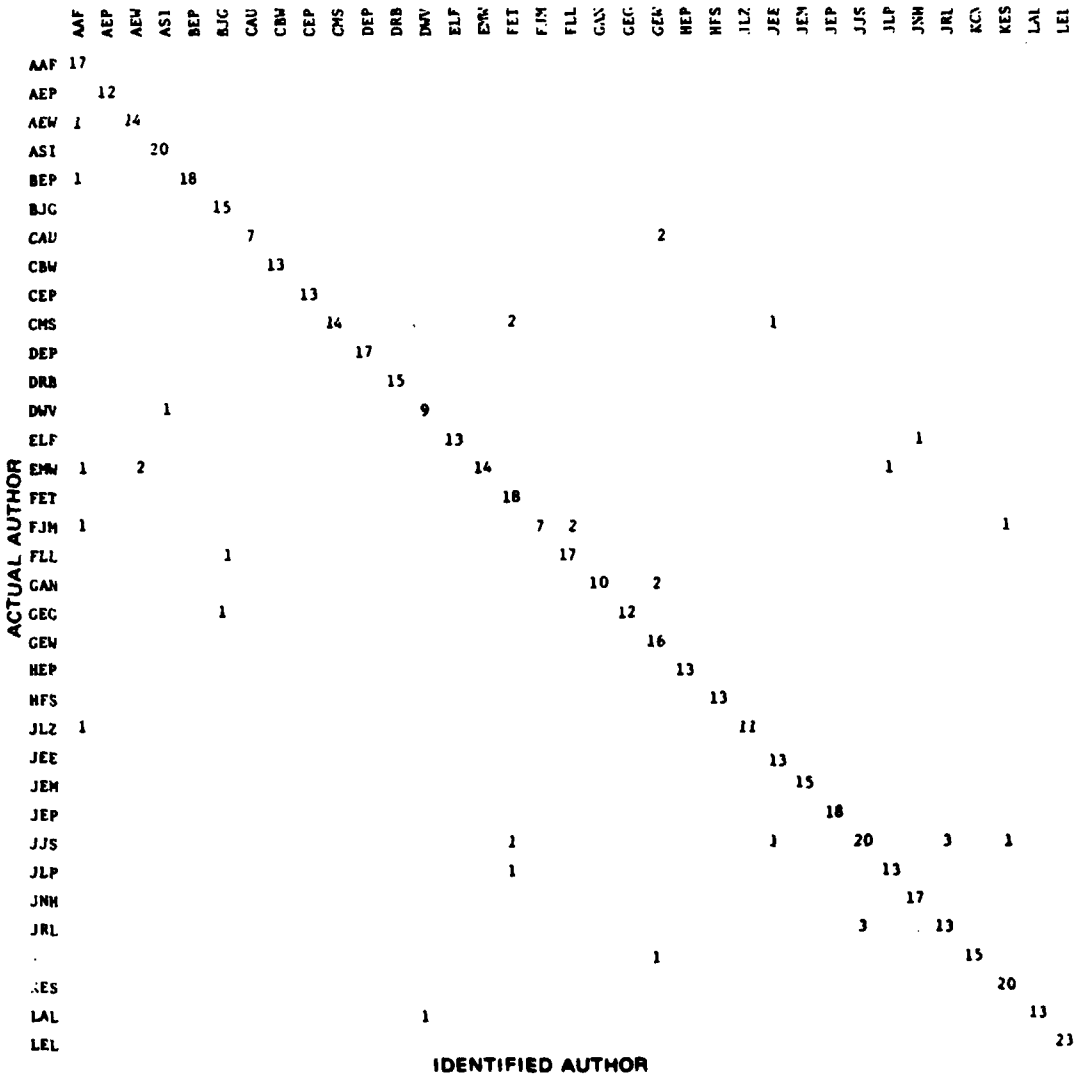


FIGURE 15 SUMMARY OF IDENTIFICATION RESULTS SUBJECT BY SUBJECT

having beem written by subject AAF. If the recognition rate were 100 percent, there would be no off-diagonal terms in Figure 15. Only about half of the subjects are listed in Figure 15 because a 59 × 59* identification table would not fit on one page. The intent, in any case was not to exhaustively list all the subject-by-subject identification results but to present an example of how the identification results were distributed.

D. Human Engineering and User Acceptance

Among the subjects polled, there were only two minor complaints concerning the signature verification system. The first of these was that it was difficult to see what one was writing because of the relatively large cylindrical structure at the writing end of the pen. The second had to do with the wire attached to the pen. However, all subjects adapted very quickly, and these problems did not affect the system's operation.

*All 59 subjects were used in this part of the analysis.

V SUMMARY

In previous sections we described the data base and data collection protocol, the signature verification algorithms and associated data base analysis procedures, and the results of the performance analysis. The performance analysis section presented estimates of the access time and Type I/Type II error curves for three signature verification algorithms* under a variety of conditions. In this section we provide a summary of the most essential results of the performance analysis.

The average access time was 8.5 seconds, dominated by the time required to write signatures. Type I/Type II error curves for the features technique using a standard feature set for all subjects was shown in Figure 11. The equal-error rate (the percent error at the point where the Type I/Type II curves intersect) is slightly more than 1 percent. These curves were calculated using attempted forgery data and all the true signatures in the data base collected with the subjects sitting at a table. The 648 attempted forgeries were obtained from trained forgers who were given copies of the true signers' signatures, instructed in how the signature verification system worked and what it measured, allowed to watch video tapes of the true signers writing their signatures, and allowed to practice as much as they desired over a three-week period. Enrollment criteria, based on the variance of the template, were imposed so that subjects who were extremely variable in writing their signatures were not accepted by the system. Of the 59 subjects in the data base, only three were unable to meet these enrollment criteria.

We believe that the Type I/Type II error curves in Figure 11 provide a realistic and probably conservative estimate (i.e., slightly worse than it really should be) of system performance for the following reasons:

- The same feature set was used for all subjects.
- Careful separation of testing and training data was always maintained.
- The analysis simulated a real-world enrollment procedure in which only a few signatures were available from which to construct the templates.
- In a real-world signature-verification application, a subject risks being denied access if he is careless or sloppy in writing his signature, but there was no comparable motivation for the subjects to cooperate in the type of data collection effort described here. In an attempt to provide at least some motivation, cash prizes were offered for the most consistent signatures, but in practice this was

*The three signature verification algorithms were a features technique based on a standard feature set (i.e., a single best set of features for all subjects collectively), a features technique based upon individualized feature sets (i.e., a best feature set derived for each subject individually), and a "rubbery" correlation algorithm.

not greatly successful. The lack of motivation led to increasing signature variances toward the end of the data collection period for most subjects, which probably caused some overestimation of the system's error rates.

Signature verification based on the features technique but with individualized feature selection was also considered. This algorithm was tested using the problem subjects (the few subjects in the data base who caused essentially all the errors) and yielded substantially improved performance compared to the features technique based on a standard feature set for all subjects (on which the results of Figure 11 are based). Although we were unable, because of time limitations, to process enough data to obtain a statistically confident estimate of the Type I/Type II error curves for individualized feature selection, based on the results of our limited testing with problem subjects and our previous experience, we believe that the equal-error rate is probably at least a factor of two better than for the features technique using a standard feature set for all subjects. The primary disadvantage of individualized feature selection is that it may require a relatively large numer of enrollment signatures.

Finally, the "rubbery" correlation algorithm was also tested using the problem subject's data. Compared to the features technique based on a standard set for all users, there was a dramatic reduction in error rate. However, as was the case for the individualized feature selection technique, because of time limitations we were unable to process enough data to provide a statistically confident estimate of the overall system Type I/Type II error curves for the rubbery correlation algorithm. This procedure used only nine signatures, comparable to that required for the standard features technique. The only disadvantage of the rubbery correlation technique is that it requires more processing time and computer storage for subject templates.

In sum, dynamic signature verification based on a three-axis pen system yields equal-error rates on the order of one percent using a features algorithm and a standard set of features for all subjects. Analysis of a limited data set indicates that a substantial reduction in error rate can be obtained by individualized feature selection or rubbery correlation algorithms, but at the cost of an increased computational burden. These are promising areas for future development.

Appendix A

DESCRIPTION OF THE MAGNETIC TAPE CONTAINING THE SIGNATURE VERIFICATION DATA BASE

DESCRIPTION OF THE MAGNETIC TAPE CONTAINING THE SIGNATURE VERIFICATION DATA BASE

A summary of the true signature, numeral, and forgery data was given in Section II. This data is stored on tape RADC SIGNATURE DATA BASE. The tape was generated using PIP (the version that supports magnetic tape reading and writing) on our PDP 11/40. It is a nine-track, 1600-CPI tape, with volume label = JSO. To read this tape on the PDP 11/70 under RSX 11-M the following steps are required:

1. Mount the tape on a 1600 CPI, nine-track tape drive.
2. Allocate MT: (MCR > ALL MT:).
3. Mount the tape (MCR > MOU MT:JSO).
4. PIP can now be used to copy the data on the magnetic tape to the system disk (or some other disk). For example, to copy file TEST.FTN to disk DRØ: use

   PIP>DRØ = MT:TEST.FTN/BS:8192.
   ↳must include period

   Note: Not all versions of PIP read from device MT: properly, so the correct version must be used.

The true signatures for a subject are stored sequentially, one signature per record, in a file of the form

TABCXABC.DAT;1

where ABC is the initials of a particular subject. Since there were 59 subjects in the data base, there are 59 such files on the tape.

Similarly, there are 59 files of numeral data for each subject. These are of the form

NNUMXABC.DAT;1

The attempted forgery data is of the form

FABCXDEF.DAT;1

where ABC are the initials of the true signer and DEF are the initials of the forger.

A test program, TEST.FTN, to read data from the signature, numeral, or forgery files and write the data out on a file TEST.LST is provided on tape RADC SIGNATURE DATA BASE. This program and an example of the programs output is given below.

As shown in the test program listing, the form of the read statement for a particular record (or signature) is

```
READ(2)  SAMPID, AUTHID, NSAMPS, ICORT, IDORW
         (IDATE(I),I=1,5),(ITIME(I),I-1,3)
         MNRESP,RMSDIF,(OLDVAL(J),J-1,44),
         ((JDATA(K,I),I=1,NSAMPS),K=1,3)
```

SAMPID is the label of the response and AUTHID is the writer identification. For example, if the record contained a true signature of subject ABC then SAMPID = ABC and AUTHID = ABC. If it is a numeral by the same subject then SAMPID = NUM and AUTHID = ABC. For the forgery files SAMPID contains the initials of the true signer and AUTHID the initials of the forger. NSAMPS is the total number of P, X, or Y samples. NSAMPS * 0.01 gives the length in seconds of the signature. The array OLDVAL contains the values of 44 features discussed in III-A-2 and DATA contains the P, X, Y data (DATA is of size 3*NSAMPS).

```
C  Program TEST.FTN
C-----------------------------------------------------------------------
C  This program reads a data file (specified by typing in the
C  subjects initials) and prints out the data in specified records on
C  file TEST.LST.  The program prompts the operator for the record
C  numbers to be printed out.  Each record contains all the data for
C  a particular true signature, numeral, or attempted forgery.  To
C  get a hardcopy of the output, spool file TEST.LST to the line
C  printer (i.e. use PIP and the /SP switch --> PIP/TEST.LST/SP)
C-----------------------------------------------------------------------
      DIMENSION OLDVAL(44),IDATE(5),ITIME(3),JDATA(3,3000)
      DIMENSION NAMDAT(9)
C-----------------------------------------------------------------------
C    Initialize a disk file to receive debugging printout
      CALL ASSIGN(6,'ZZ0:TEST.LST;1',14)
C-----------------------------------------------------------------------
C    Get the initials of the subject
      TYPE 5
    5 FORMAT(/'$ True signature, Numeral, or Forgery data'
     *       ,' (type T,N, or F) ')
      ACCEPT 7, ICHECK
    7 FORMAT(A1)
      TYPE 10
   10 FORMAT(/'$ Initials of the true signer (3 characters) ')
      ACCEPT 15,AUTHOR
   15 FORMAT(A3)
      IF(ICHECK .EQ. 'F') GO TO 22
      IF(ICHECK .EQ. 'T') ENCODE(18,20,NAMDAT) AUTHOR,AUTHOR
      IF(ICHECK .EQ. 'N') ENCODE(18,21,NAMDAT) AUTHOR
   20 FORMAT('ZZ0:T',A3,'X',A3,'.DAT;1')
   21 FORMAT('ZZ0:NNUM','X',A3,'.DAT;1')
      GO TO 27
   22 TYPE 23
   23 FORMAT(/'$ Initials of the Forger (3 characters) ')
      ACCEPT 24, FORGID
   24 FORMAT(A3)
      ENCODE(18,25,NAMDAT) AUTHOR,FORGID
   25 FORMAT('ZZ0:F',A3,'X',A3,'.DAT;1')
   27 TYPE 28, NAMDAT
   28 FORMAT(/'  The file opened is ',9A2)
   24 CALL ASSIGN(2,NAMDAT,18)
C-----------------------------------------------------------------------
C    Get the number of the response to be printed or plotted
   29 REWIND 2
      TYPE 30
   30 FORMAT(/'$ Record number (integer) = ')
      ACCEPT 31,NUMREC
   31 FORMAT(I3)
C    Skip records up to the one specified
      IF(NUMREC .EQ. 1)  GO TO 37
      DO 36 I=1,NUMREC-1
   36   READ(2,END=100)
C-----------------------------------------------------------------------
C    Read in the data from the specified record
   37 IF(ICHECK .EQ. 'F')  GO TO 39
      READ(2)  SAMPID,AUTHID,NSAMPS,ICONT,IDONW
     *        ,((IDATE(I), I=1,5),(ITIME(I), I=1,3),MNRESP,MNSDIF
     *        ,(OLDVAL(J), J=1,44),((JDATA(K,I),I=1,NSAMPS),K=1,3)
      GO TO 50
   39 READ(2)  SAMPID,AUTHID,NSAMPS,IDED
     *        ,((IDATE(I), I=1,5),(ITIME(I), I=1,3),MNRESP,MNSDIF
     *        ,(OLDVAL(J), J=1,44),((JDATA(K,I),I=1,NSAMPS),K=1,3)
C     Output data values
   50 WRITE(6,52) SAMPID,AUTHID,NSAMPS,MNRESP,MNSDIF,IDATE,ITIME
      WRITE(6,54)
      WRITE(6,55) (I,OLDVAL(I), I=1,44)
      WRITE(6,58)
      DO 51 I=1,NSAMPS
        WRITE(6,59) I,JDATA(1,I),JDATA(2,I),JDATA(3,I)
   51 CONTINUE
   52 FORMAT(1H1,5X,' SAMPID = 'A4,3X,'AUTHID = 'A4,3X,'NSAMPS = '14
     *       ,3X,'MNRESP = '13.3X,'MNSDIF = 'F6.2
     *       ,//,5X,'  DATE = '5A2,5X,'TIME = '5A2)
   54 FORMAT(///23X,'I',7X,'Feature Value'/)
   55 FORMAT(20X,I4,6X,F10.3)
   58 FORMAT(1H1,23X,'I',8X,'X values',4X,'Y values'
     *       ,5X,'P values'/)
   59 FORMAT(21X,I4,8X,I5,7X,I5,8X,I5)
C-----------------------------------------------------------------------
C    Repeat if desired
C    Allow the option to exit or to specify another record
   70 TYPE 75
   75 FORMAT(/'$ Print out another record? (Y for Yes or N for No)')
      ACCEPT 76,ICH
   76 FORMAT(A1)
      IF(ICH .EQ. 'N') GO TO 9000
      IF(ICH .EQ. 'Y') GO TO 29
      GO TO 70
C-----------------------------------------------------------------------
  100 TYPE 101
  101 FORMAT(//'  **** ERROR  --  Record number out of range ****'//)
      GO TO 29
C-----------------------------------------------------------------------
 9000 CONTINUE
      END
```

THE DATA AND PROGRAM OUTPUT IS ASSIGNED TO A PSEUDO DEVICE ZZO:, BEFORE EXECUTING THE PROGRAM, THIS PSEUDO DEVICE MUST BE ASSIGNED TO THE ACTUAL DISK ON WHICH THE DATA IS STORED. FOR EXAMPLE, IF THE DATA IS STORED ON DR1: THEN MCR> ASN DR1: = ZZO:

FIGURE A-1 TEST PROGRAM LISTING

```
$ASN DR1:=Z20
$RUN TEST

 True signature, Numeral, or Forgery data (type T,N, or F) T

 Initials of the true signer (3 characters) CMS

 The file opened is Z20:TCMSXCMS.DAT;1

 Record number (integer) = 3

 Print out another record? (Y for Yes or N for No)
```

FIGURE A-2 EXAMPLE OF TEST PROGRAM EXECUTION

SAMPID = CMS AUTHID = CMS NSAMPS = 672 NMRESP = 3 RMSDIF = 1.73
DATE = 02-JUN-88 TIME = 12:54:

I	Feature Value
1	-2.309
2	26.963
3	17.886
4	-24.024
5	8.259
6	6.867
7	386.000
8	-10.019
9	263.000
10	48.000
11	-6.955
12	37.471
13	16.829
14	-29.073
15	8.439
16	7.095
17	385.000
18	-10.111
19	264.000
20	50.000
21	24.462
22	37.629
23	21.513
24	-24.462
25	8.038
26	7.135
27	346.000
28	-8.830
29	303.000
30	42.000
31	41.910
32	45.902
33	45.975
34	20.195
35	23.784
36	-2.948
37	21.715
38	22.118
39	48.923
40	6.650
41	5.000
42	0.580
43	6.000
44	6.070

NOTE: Program TEST writes this data out to file TEST.LST;1 instead of directly to the line printer.

I	X values	Y values	P values
1	-1	1	1
2	-1	0	3
3	-1	3	6
4	-1	4	11
5	-3	9	15
6	-5	11	21
7	-5	13	29
8	-8	15	35
9	-10	18	43
10	-11	21	49
11	-14	24	57
12	-14	24	65
13	-21	13	71
14	-30	-9	75
15	-42	-32	81
16	-51	-47	86
17	-56	-59	89
18	-58	-65	90
19	-61	-70	91
20	-61	-77	92
21	-61	-79	95
22	-51	-82	99
23	-40	-77	102
24	-21	-64	107
25	-5	-46	106
26	3	-30	104
27	11	-18	102
28	14	-8	98
29	16	-3	93
30	16	5	89
31	14	12	86

FIGURE A-3 EXAMPLE OF OUTPUT OF PROGRAM TEST

32	10	16	84	119	13	-43	94
33	3	19	83	120	17	-35	89
34	-9	18	81	121	17	-24	86
35	-24	11	78	122	20	-16	85
36	-35	3	77	123	20	-9	81
37	-41	-9	78	124	20	-2	78
38	-48	-17	78	125	17	3	72
39	-52	-25	79	126	12	7	71
40	-53	-34	79	127	11	8	71
41	-56	-39	82	128	10	12	71
42	-58	-51	87	129	4	11	72
43	-61	-60	90	130	-2	9	73
44	-61	-68	95				
45	-61	-76	95	132	-23	-12	80
46	-60	-84	98	133	-28	-24	86
47	-54	-92	103	134	-32	-39	91
48	-44	-102	106	135	-34	-52	95
49	-29	-103	113	136	-37	-61	100
50	-11	-95	119	137	-37	-72	107
51	1	-82	122	138	-36	-82	112
52	15	-69	123	139	-33	-89	114
53	18	-56	116	140	-24	-90	115
54	20	-45	102	141	-15	-84	117
55	15	-32	82	142	-1	-74	118
56	8	-21	57	143	9	-63	119
57	7	-14	42	144	17	-48	116
58	4	-11	28	145	20	-32	114
59	4	-6	21	146	20	-19	106
60	3	-6	13	147	22	-11	99
61	0	-3	9	148	21	-4	90
62	1	-3	8	149	17	1	79
63	0	-3	6	150	14	4	65
64	1	-2	4	151	9	8	55
65	0	-1	2	152	8	8	48
66	-1	2	1	153	6	5	44
67	-2	-1	2	154	7	0	43
68	1	1	2	155	7	-10	46
69	0	-1	2	156	6	-19	53
70	0	-1	0	157	6	-31	65
71	-2	1	1	158	5	-41	75
72	-2	-3	8	159	4	-52	86
73	-4	-7	15	160	0	-62	94
74	0	-9	26	161	-6	-68	102
75	4	-11	38	162	-13	-71	110
76	7	-11	49	163	-19	-72	114
77	13	-11	62	164	-29	-71	115
78	17	-9	72	165	-38	-72	115
79	20	-7	79	166	-45	-69	116
80	23	-5	83	167	-45	-70	118
81	22	-2	85	168	-42	-74	118
82	24	-2	84	169	-35	-76	117
83	23	-4	80	170	-23	-76	114
84	23	-5	78	171	-11	-68	113
85	21	-8	73	172	0	-60	110
86	17	-12	69	173	8	-48	107
87	13	-22	68	174	11	-42	104
88	7	-29	65	175	15	-32	97
89	1	-33	63	176	16	-22	93
90	-5	-34	60	177	18	-13	89
91	-10	-34	59	178	17	-2	85
92	-8	-34	62	179	13	6	82
93	-4	-29	65	180	10	12	74
94	0	-24	69	181	3	17	70
95	4	-15	72	182	-4	15	70
96	7	-4	74	183	-11	8	71
97	9	-1	78	184	-18	-10	75
98	8	5	79	185	-25	-32	82
99	3	8	82	186	-30	-48	92
100	-3	13	80	187	-28	-68	104
101	-13	14	79	188	-24	-78	115
102	-24	9	81	189	-14	-85	127
103	-30	5	81	190	-2	-83	130
104	-35	-1	81	191	6	-73	133
105	-41	-9	80	192	12	-62	130
106	-46	-13	79	193	15	-44	122
107	-48	-23	82	194	14	-23	110
108	-51	-29	84	195	10	-4	98
109	-52	-37	86	196	5	7	84
110	-50	-42	86	197	3	12	74
111	-50	-47	88	198	1	13	61
112	-47	-53	91	199	-2	13	52
113	-37	-58	92				
114	-28	-63	94	201	-12	4	41
115	-18	-63	95	202	-13	-8	47
116	-10	-60	95	203	-14	-24	54
117	1	-55	97	204	-11	-39	64
118	9	-49	96	205	-8	-58	74

FIGURE A-3 EXAMPLE OF OUTPUT OF PROGRAM TEST (CONTINUED)

206	-4	-59	85	293	-35	-1	136
207	4	-63	99	294	-24	17	139
208	11	-61	110	295	-11	31	139
209	19	-56	120	296	-2	37	134
210	28	-46	128	297	6	36	127
211	29	-38	130	298	11	36	119
212	35	-29	134	299	12	33	107
213	35	-19	130	300	15	27	93
214	35	-13	127	301	17	24	85
215	34	-7	118	302	19	16	80
216	31	-2	113	303	18	9	77
217	30	1	109	304	17	-2	76
218	27	6	103	305	15	-14	77
219	24	13	97	306	8	-29	80
220	19	18	88	307	-2	-46	86
221	12	24	82	308	-14	-56	89
222	3	19	81	309	-25	-66	92
223	-10	7	80	310	-37	-69	96
224	-23	-8	83	311	-49	-68	101
225	-33	-24	87	312	-56	-69	106
226	-45	-38	93	313	-62	-67	111
227	-54	-54	101	314	-67	-63	118
228	-61	-68	109	315	-64	-52	126
229	-67	-85	118	316	-54	-32	134
230	-70	-100	124	317	-35	-10	144
231	-69	-111	131	318	-19	5	149
232	-61	-121	139	319	-3	17	153
233	-45	-125	143	320	7	24	150
234	-27	-124	147	321	12	25	141
235	-7	-114	148	322	18	23	136
236	5	-97	146	323	17	21	114
237	17	-79	139	324	18	19	98
238	21	-60	126	325	16	17	89
239	21	-43	106	326	19	15	84
240	16	-29	80	327	20	2	86
241	9	-19	54	328	15	-17	86
242	6	-13	40	329	7	-37	89
243	4	-8	29	330	-8	-52	91
244	3	-7	20	331	-24	-62	94
245	1	-5	14	332	-35	-75	99
246	0	-2	7	333	-46	-82	106
247	1	-3	8	334	-54	-92	114
248	1	-2	5	335	-61	-100	119
249	1	-1	5	336	-61	-107	128
250	1	-1	3	[illegible]			
251	0	1	1	338	-40	-113	139
252	1	-1	2	339	-23	-106	138
253	0	0	3	340	-10	-89	127
254	1	-1	3	341	-2	-67	105
255	0	-1	1	342	1	-46	78
256	-1	1	1	343	0	-31	53
257	0	-1	1	344	1	-20	38
258	1	-1	1	345	0	-14	26
259	0	-1	2	346	-2	-8	18
260	1	-1	1	347	0	-7	14
261	-1	1	0	348	0	-5	9
262	2	-1	11	349	0	-4	8
263	5	-1	23	350	0	-1	5
264	7	-4	35	351	-2	0	4
265	8	-5	46	352	0	-2	3
266	9	0	57	353	-1	-1	2
267	11	3	72	354	0	-2	1
[illegible]	[illegible]	[illegible]	[illegible]	355	0	0	1
[illegible]	[illegible]	[illegible]	[illegible]	356	-1	1	1
270	16	12	92	357	0	-1	3
271	16	14	94	358	-1	-1	1
272	19	11	96	359	0	-1	2
273	24	12	98	360	-2	0	1
274	24	11	97	361	-1	1	1
275	25	7	95	362	0	-1	2
276	23	0	94	363	-1	0	2
277	23	-12	96	364	1	-1	2
278	21	-25	97	365	-1	-1	1
279	14	-41	95	366	-1	2	0
280	6	-55	94	367	1	-4	11
281	-7	-63	94	368	1	-11	20
282	-18	-76	98	369	3	-20	32
283	-29	-83	99	370	5	-25	44
284	-36	-85	105	371	5	-25	59
285	-47	-88	106	372	11	-26	74
286	-54	-86	108	373	16	-20	88
287	-58	-87	113	374	23	-17	98
288	-62	-84	114	375	26	-13	104
289	-65	-80	117	376	29	-8	109
290	-64	-71	119	377	30	-4	114
291	-60	-55	123	378	33	0	118
292	-49	-31	130	379	34	5	118

FIGURE A-3 EXAMPLE OF OUTPUT OF PROGRAM TEST (CONTINUED)

380	34	9	116	467	31	-34	126
381	32	17	114	468	36	-21	133
382	29	19	109	469	34	-8	138
383	22	23	100	470	36	3	141
384	18	26	90	471	36	12	142
385	10	29	82	472	35	13	142
386	-1	30	76	473	32	19	139
387	-8	21	74	[illegible]	[illegible]	[illegible]	[illegible]
388	-15	10	74	475	26	32	121
389	-22	-5	77	476	19	34	109
390	-25	-23	82	477	17	33	100
391	-26	-41	88	478	13	31	92
392	-22	-62	99	479	12	30	90
393	-18	-81	110	480	10	26	90
394	-14	-97	122	481	4	18	94
395	-12	-111	134	482	-1	3	98
396	-12	-125	143	483	-11	-18	100
397	-15	-133	153	484	-18	-37	103
398	-18	-135	158	485	-25	-52	105
399	-25	-132	160	486	-29	-64	107
400	-36	-119	157	487	-31	-76	110
401	-53	-98	150	488	-33	-85	112
402	-64	-72	143	489	-36	-90	115
403	-68	-49	131	490	-37	-93	115
404	-66	-32	118	491	-38	-95	116
405	-60	-20	102	492	-36	-97	118
[illegible]	[illegible]	[illegible]	[illegible]	493	-33	-98	117
407	-46	-13	81	494	-30	-98	116
408	-36	-19	75	495	-25	-96	114
409	-23	-29	78	496	-18	-92	113
410	-8	-33	79	497	-5	-83	119
411	2	-35	83	498	5	-72	123
412	11	-36	93	499	13	-59	126
413	17	-34	102	500	18	-41	124
414	23	-31	111	501	20	-26	122
415	28	-29	120	502	22	-14	119
416	32	-23	128	503	23	-4	117
417	36	-19	135	504	22	2	110
418	38	-8	142	505	22	8	104
419	36	-1	142	506	20	11	100
420	33	11	141	507	23	6	100
421	27	28	137	508	25	-5	100
422	25	27	131	509	25	-19	106
423	28	33	122	510	24	-38	110
424	17	36	111	511	19	-54	114
425	13	36	100	512	12	-68	116
426	9	37	90	513	2	-72	116
427	8	32	90	514	-11	-71	114
428	8	28	89	515	-26	-66	109
429	6	20	94	516	-40	-58	104
430	3	8	96	517	-47	-39	102
431	-5	-13	102	518	-49	-27	95
432	-14	-38	108	519	-49	-15	91
433	-21	-56	113	520	-48	-6	85
434	-28	-67	118	521	-43	-3	82
435	-33	-72	119	522	-36	-4	80
436	-39	-77	120	523	-26	-10	79
437	-43	-84	123	524	-13	-18	83
438	-48	-88	124	525	-3	-24	87
439	-52	-91	125	526	7	-29	92
440	-51	-92	124	527	17	-31	100
441	-47	-97	123	528	20	-33	106
442	-36	-99	127	529	25	-30	112
443	-26	-99	130	530	31	-26	118
444	-12	-96	133	531	33	-20	121
445	3	-87	136	532	36	-10	126
446	13	-75	136	533	35	-8	128
447	22	-65	140	534	33	5	129
448	27	-52	138	535	28	14	126
449	30	-38	135	536	21	23	123
450	31	-22	127	537	17	28	118
451	28	-7	119	538	10	31	115
452	22	4	110	539	3	28	109
453	18	13	98	540	-6	24	103
454	11	17	86	541	-16	11	99
455	3	19	76	542	-24	-9	98
456	-5	16	69	[illegible]	[illegible]	[illegible]	[illegible]
457	-11	4	64	544	-31	-40	102
458	-15	-9	65	545	-32	-51	107
459	-20	-23	67	546	-30	-65	109
460	-22	-35	70	547	-23	-73	110
461	-22	-43	74	548	-13	-81	125
462	-14	-53	81	549	-4	-84	132
463	-6	-58	87	550	8	-80	137
464	3	-58	97	551	18	-71	142
465	16	-54	106	552	26	-60	147
466	22	-47	115	553	31	-44	147

FIGURE A-3 EXAMPLE OF OUTPUT OF PROGRAM TEST (CONTINUED)

554	33	-26	144	614	3	36	142
555	30	-6	139	615	-11	35	122
556	23	6	134	616	-17	28	100
557	17	17	127	617	-17	19	78
558	7	26	118	618	-13	13	56
559	-3	27	105	619	-9	8	40
560	-11	28	93	620	-6	6	27
561	-17	24	82	621	-5	5	19
562	-20	15	74	622	-3	2	15
563	-21	9	64	623	-2	1	10
564	-23	1	62	624	-1	0	8
565	-22	-7	59	625	-1	1	5
566	-20	-15	61	626	-2	2	4
567	-14	-26	64	627	0	1	4
568	-10	-35	68	628	-2	0	3
569	-1	-39	73	629	-1	-1	3
570	4	-39	76	630	-1	-1	6
571	9	-33	82	631	-6	-2	19
572	15	-30	87	632	-12	-11	34
573	19	-22	90	633	-22	-16	51
574	21	-17	90	634	-37	-20	70
575	21	-10	87	635	-50	-22	87
576	21	-5	86	636	-65	-24	105
577	23	-4	86	637	-69	-27	119
578	23	-6	86	638	-69	-26	121
579	23	-12	88	639	-64	-27	120
580	24	-18	92	640	-59	-22	111
581	20	-34	101	641	-53	-18	99
582	15	-53	113	642	-41	-16	84
583	5	-69	122	643	-31	-11	64
584	-8	-80	133	644	-20	-8	44
585	-21	-81	142	645	-13	-5	30
586	-29	-81	151	646	-11	-3	21
587	-26	-71	162	647	-6	-1	16
588	-13	-53	169	648	-5	-2	11
589	4	-33	173	649	-4	-2	9
590	16	-17	169	650	-2	1	5
591	20	-9	163	651	-4	1	4
592	28	-2	157	652	-2	0	12
593	28	0	148	653	-12	0	29
594	28	3	138	654	-24	-7	50
595	27	5	128	655	-40	-13	71
596	25	3	126	656	-53	-17	98
597	20	-12	123	657	-64	-21	124
598	8	-34	122	658	-71	-23	143
599	-3	-52	124	659	-69	-22	153
600	-13	-66	124	660	-65	-22	146
601	-21	-73	127	661	-57	-21	128
602	-20	-79	129	662	-44	-20	104
603	-13	-81	131	663	-32	-14	76
604	3	-76	136	664	-21	-11	52
605	15	-66	141	665	-14	-6	35
606	24	-54	146	666	-10	-3	24
607	31	-43	153	667	-6	-4	18
608	39	-26	159	668	-3	-2	14
609	42	-15	163	669	-3	-3	10
610	40	-6	164	670	-2	-1	5
611	36	0	167	671	-2	2	4
612	30	9	166	672	0	0	4
613	18	28	155				

FIGURE A-3 EXAMPLE OF OUTPUT OF PROGRAM TEST (CONCLUDED)

Appendix B

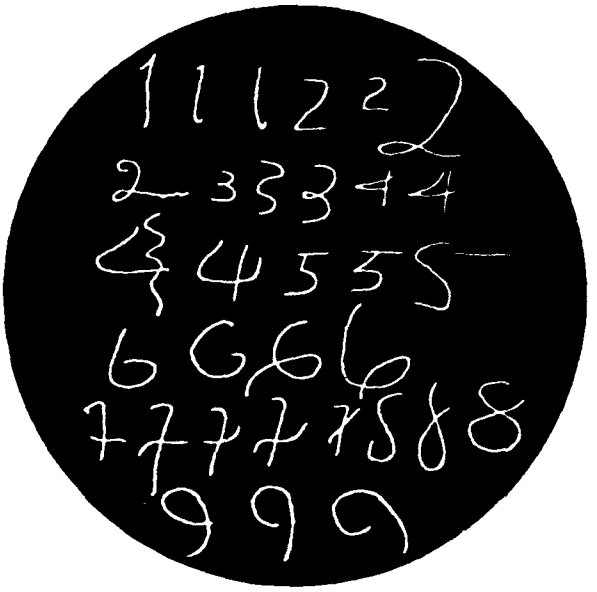
AN ON-LINE DATA ENTRY SYSTEM FOR HAND-PRINTED CHARACTERS

Special Feature:

An On-Line Data Entry System for Hand-Printed Characters*

H. D. Crane
Stanford Research Institute

R. E. Savoie
Telesensory Systems, Inc.



Introduction

The primary method of entering large amounts of routinely produced, hand-printed data into computer systems is via manual keyboards. Manual retranscription, however, entails a number of disadvantages such as extra cost, delays, and errors.

Optical character recognition attempts to bypass the manual retranscription process by providing automatic reading of source documents. However, since OCR processing typically is separate from document origination, the generator of the document cannot realize the benefits that accrue to real-time, on-line automated data entry. Often there is no way of knowing when substitution errors have occurred, and OCR equipment is costly relative to other methods.

Real-time character recognition, i.e., capturing the material as it is written, obviates the need for manual retranscription or OCR, and provides for immediate error detection and correction. However, a keyboard that accommodates a large character set — plus a hard-copy printer for each data entry station — can be quite bulky and expensive.

Alternatively, a direct entry system may use an inexpensive writing device to make its own hard copy and to produce machine-recognizable code. Writing systems to track pen motions have been previously described, but such systems require special writing surfaces[1] or special writing environments.[2] Therefore these systems, like the keyboard printer, also tend to be bulky and expensive.

This paper describes a system that uses a specially instrumented ball-point pen requiring no special writing surface. Unlike many OCR techniques, the method described is dynamic. That is, instead of a post facto analysis of a complete input pattern — e.g., in terms of loops, corners, and height — the character recognition is based on real-time detection and analysis of the sequence of writing directions taken by the pen. Each character is described in terms of an allowed set of stroke direction sequences. The character actually recognized by the system can be echoed to be verified immediately by the person generating the document.

Monitoring the direction of motion

The writing system is based on the three-dimensional force generated at the pen tip during writing. This force consists of the downward force directed toward the paper and the drag force in the plane of the paper.

From force measurements alone it is not possible to derive an accurate measure of pen velocity (and therefore of pen position), because drag varies with paper friction, the exact orientation of the pen, and pen pressure. Furthermore, the system has no knowledge of pen motion when the pen is lifted from the paper. However, absolute pen position (although necessary for entering pictorial input material or for reconstructing the exact form of each input character as drawn) is *not* necessary for character recognition. It is sufficient, as we show subsequently, to determine the *sequence* of direction movements, which is readily obtained from the force measurement.

The force-measuring instrumentation is incorporated into the pen tip without any instrumentation of the writing surface or the writing area. The vertical force on the paper indicates when the pen is "down" or "up," i.e., on or off the paper. The instantaneous direction of pen motion is readily determined from the lateral forces in the plane of the paper.

Three-dimensional force-sensitive pen

A previous article[3] on a direction-sensitive pen and its potential use in hand-printed character recognition showed that English letters can be described by a sequence of connected up/down and left-right movements. The pen used a pivoted writing shaft that moved in response to the writing force and made electrical contact with one of four segments of a commutator ring. Although the device showed the feasibility of such an instrument, it was crude and unreliable because it required mechanical

*This system was conceived and developed by the authors at Stanford Research Institute. Xebec Systems, Inc., Santa Clara, California, is developing a commercial version of the system under license from SRI.

Reprinted from *Computer Magazine*, Vol. 10, No. 3, pp. 43-50 (March 1977).

motion of the entire writing shaft. A later version used a light-emitting diode at the upper end of the pivoted writing shaft; the light was directed toward a stationary quadrant photocell. The position of the shaft was tracked by monitoring the movement of the light with respect to the photocell. This optical system provided better force sensitivity than the earlier version, but still required the entire writing shaft to move. Also, both of these systems required additional measurements to determine vertical pressure.

In the most recent design, the pen point is mounted to a diaphragm containing a system of strain gauges to detect the instantaneous lateral and vertical forces on the point. This version is shown in Figure 1. The pen point must be maintained in a nominally vertical direction during writing; the angle of the barrel can be adjusted to suit the individual user.

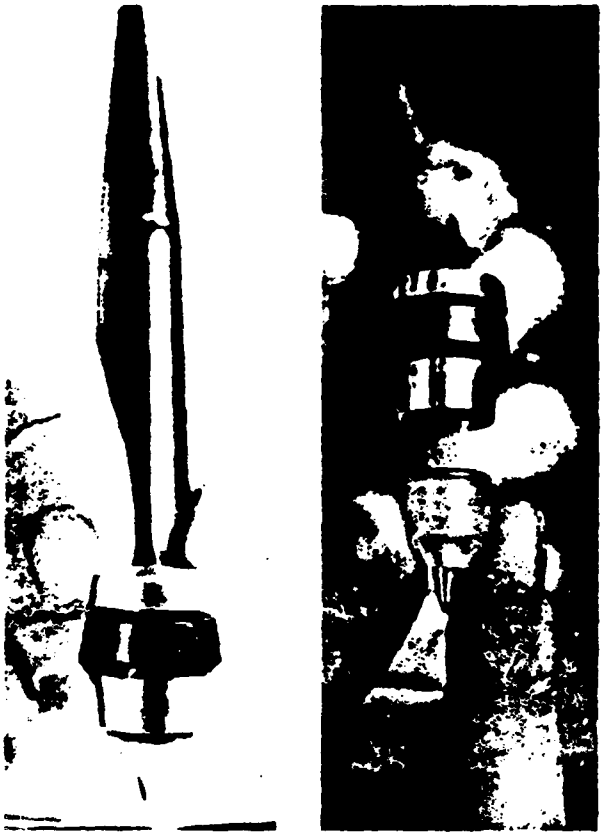
Figure 2 shows the photographically etched strain-gauge array, which is bonded to the diaphragm inside the housing that holds the replaceable ink cartridge. The center of the diaphragm is rigidly connected to the pen body, as shown in Figure 3a. The force generated at the writing tip distorts the diaphragm, as shown exaggerated in Figures 3b and 3c. With normal writing, the pen point deflects less than a thousandth of an inch.

It is easier to describe the operation of the strain-gauge system if we imagine that the eight gauges are arranged in four pairs, as shown in Figure 3a, rather than in the actual planar array form of Figure 2. These pairs are connected electrically in a compound bridge circuit (Figure 4) that isolates the three components of the applied force. To see how the bridge operates, let X and Y represent the left/right and near/far directions in the plane of the writing surface, and P the vertically directed force. A vertically directed force will cause the diaphragm to bend as shown in Figure 3b. The four gauges on the top of the diaphragm will be in compression, and the four gauges on the bottom of the diaphragm will be in tension. Hence, the voltages at Points A and B in the upper bridge will change by the same amount and in the same direction; these changes will cancel in the differential amplifier in the X channel. The voltages at Points C and D in the lower bridge will also change by the same amount and in the same direction, so there will be no change in the Y output either. However, the polarity of change at points C and D is opposite to that at Points A and B. Accordingly, the changes at all four points are additive in the central amplifier which measures vertical pressure. Thus, vertical force is monitored by the central channel, with no first-order coupling to the X and Y channels.

A lateral force in the X direction will cause the diaphragm to bend as shown in Figure 3c. In this case, Points A and B will move in equal but opposite directions. These changes are additive in the output of the X channel but cancel in the P channel. Thus, an X-directed force will cause a change only in the X channel. Similarly, a Y-directed force will cause a change only in the Y channel. An arbitrary force on the pen point can thus be resolved into X, Y, and P components.

Note in Figures 3b and 3c that the polarity of strain on the lower side of the diaphragm near the center is the same as the polarity of strain on the upper side of the diaphragm near the periphery. It is for this reason that the four-pair gauge system can be realized in the single-sided, planar array shown in Figure 2.

From the X and Y components of force, it is straightforward to determine the instantaneous angle of force in the plane of the writing surface (i.e., the direction of writing), as well as the magnitude of the force in that



(a) (b)

Figure 1. (a) Ball-point pen that measures the three-dimensional force generated at the tip during writing; (b) Replaceable ink cartridge and ballpoint-tip assembly.

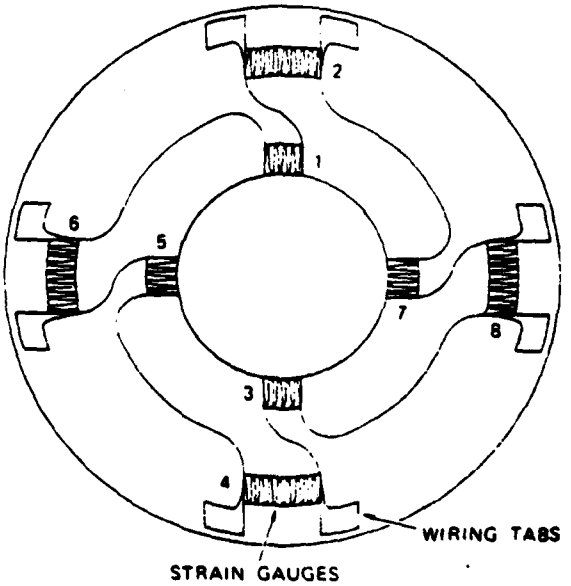


Figure 2. Schematic drawing of the photographically etched array of strain gauges that is bonded to a diaphragm inside the cartridge housing.

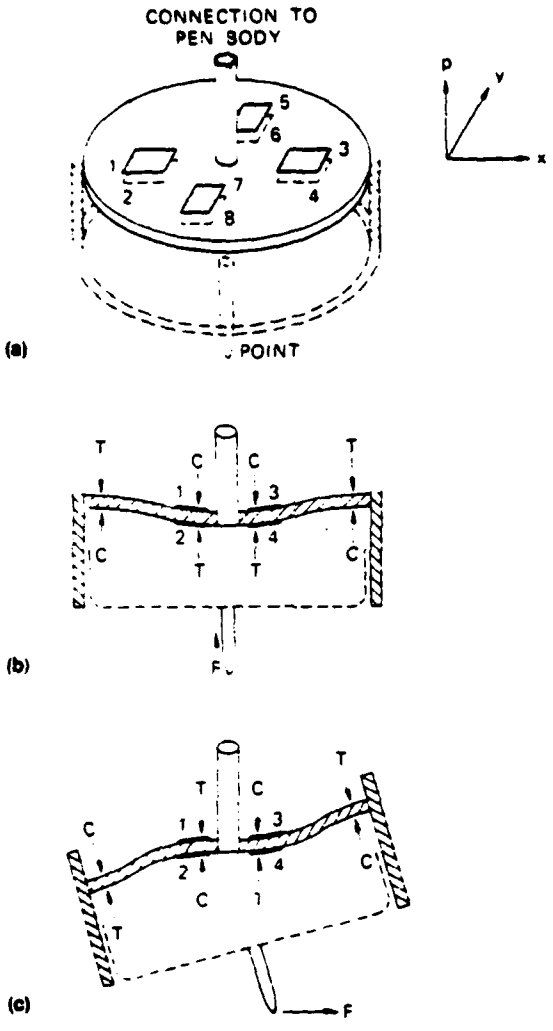


Figure 3. (a) A strain-gauge arrangement in which the paired gauges are located on opposite sides of the diaphragm; (b and c) illustration of the effect of a downward and lateral force (highly magnified).

direction. The system also provides a continuous measure of P, the vertical force (orthogonal to the writing surface). Although the pen provides high resolution force measurements, it is sufficient for hand-printed data entry to quantize the measurements quite coarsely. In the vertical direction, it is necessary to know only that the pen point is "up" or "down," i.e., when the vertical force is greater than some threshold. The X and Y signals are quantized into the four quadrant directions: up, right, down, and left, symbolized by U, R, D, and L. "Pen-up," symbolized by a ".", can be thought of as a fifth direction of motion.

The following section shows how these five direction signals (U, R, D, L, and .) can be utilized in a practical character-recognition system. In this system, the direction of writing is sampled at a clock rate of approximately 50 to 100 per second. At this clock rate, each new direction signal generally persists for many clock cycles.

Sequential character recognition algorithm

With the signals provided by the pen, direction of writing is the only information available for character recognition. As each character is printed, the pen generates a sequence

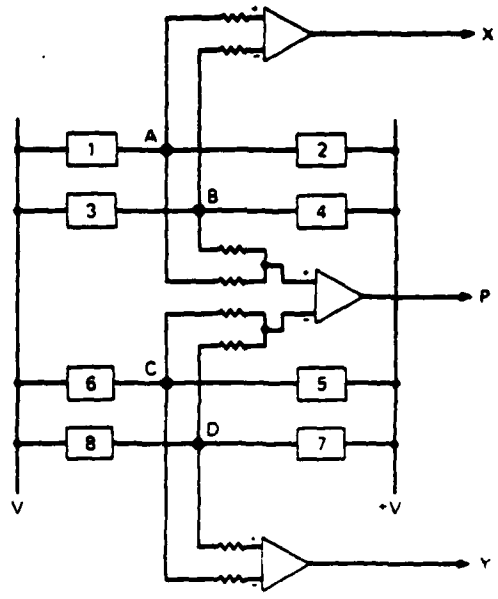


Figure 4. Compound bridge for isolating the three-dimensional force components.

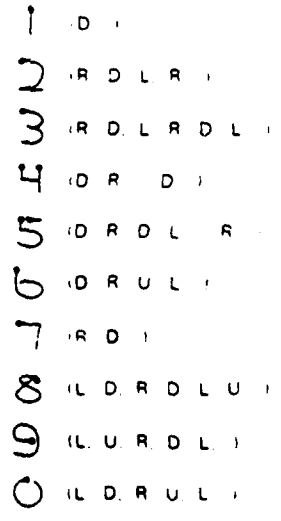


Figure 5. An idealized set of numeric characters. The symbols U,R,D,L,. represent up, right, down, left, and pen-up, respectively.

of direction signals describing its motion. With a reasonable set of constraints on character formation, the direction sequences are sufficient for machine recognition of the printed characters. Figure 5 shows a typical set of direction sequences that is unique for the ten digits. For example, the sequence for a 1 is (D,.), meaning a down stroke followed by a pen-up. It would be trivial to design a logic system to recognize each character as shown. However, there is wide variation in the way people form characters. It is advantageous, therefore, to allow as broad a range of sequences as possible for each character.

One possible approach to the sequence-recognition problem is the "table look-up," which lists all allowed sequen-

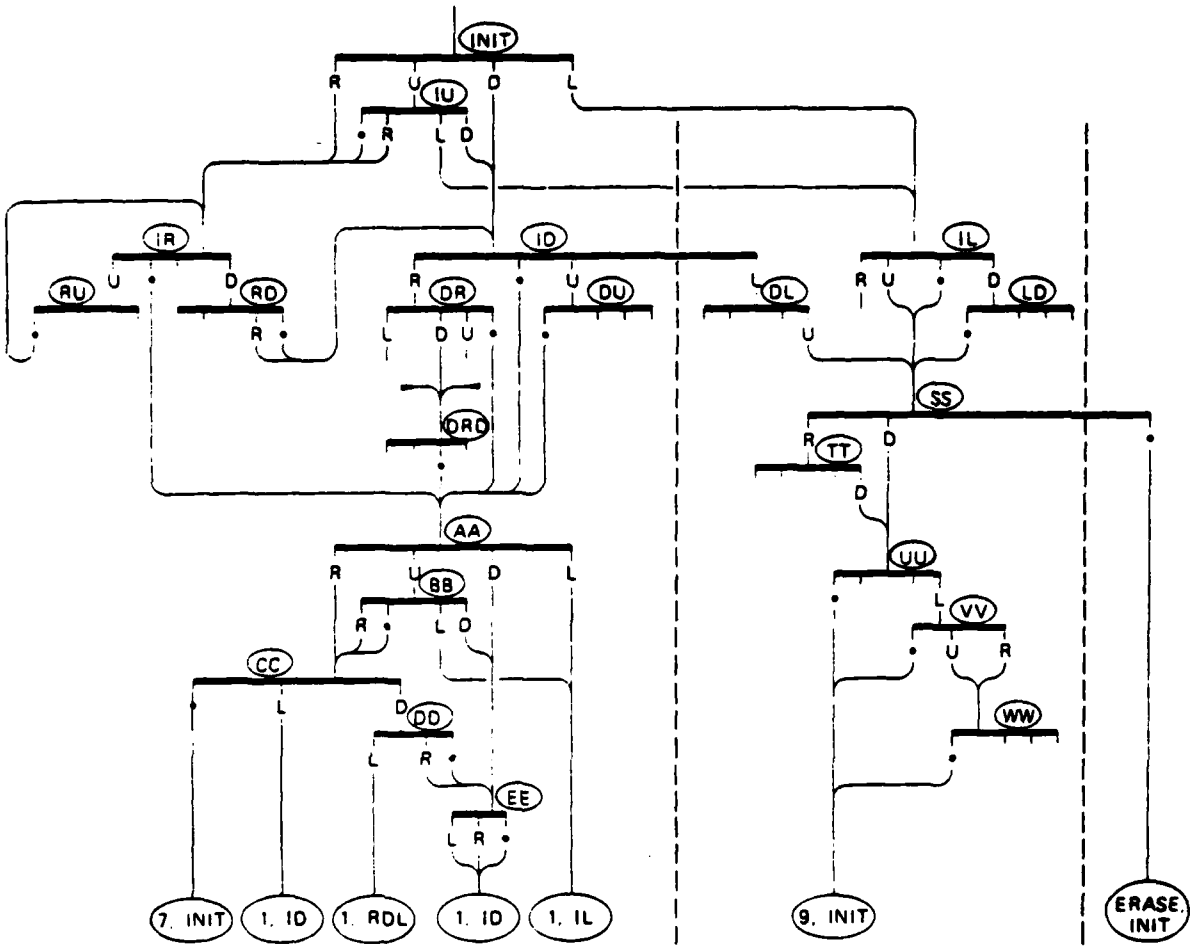


Figure 6. A portion of the state machine logic associated with the 1, 7, 9, and erase characters.

ces for each character; when a character is written, the generated sequence is compared with each entry in the table. Allowance for a wide variation in writing styles may require an excessively long table. A more efficient way utilizes a state machine with state transitions determined by the direction sequences.[4] By appropriately specifying the state transitions, particular directions or direction sequences may be ignored if they are not relevant to the recognition process.

In this section, we consider the operation of such a machine. The next section shows how the state machine can be efficiently implemented with ROM components.

Figure 6 illustrates the portion of the graph of the sequential decision machine that recognizes the digits 1, 7, and 9, as well as the "erase" character. The design demonstrates the range of possibilities that may be achievable. The figure shows broad horizontal lines, which represent the various states of the machine, and vertical LINK PATHS, which describe the state transitions. The ovals at the bottom indicate output characters and the next state following the output. The logic structure shown is that of a class-4 state machine,[5] in which both the next state, g(X,Q), and the output, f(X,Q), are determined by the present state, X, and the inputs, Q. The highest state of the machine, marked INIT (initial), becomes energized whenever a character has been recognized, and a new search begins. It is convenient to think of a marker advancing through the graph as each different direction is recognized in sequence. For example, an initial left stroke would move the marker to state IL (*I*nitial *L*eft). The marker would remain at State IL for as long as the sampled direction signal remained unchanged. If the writing subsequently turned down, D, the marker would advance to state LD (Initial *L*eft followed by *D*own). Because the state-transitions depend solely on the direction sequences, the path through the graph is independent of both speed of writing and size of characters. Exceptions are the front-end and back-end timing delays described below.

In a general state machine, any number of link paths may leave a state. Each state of the pen machine has six link paths, five corresponding to the five directions (U,R,D,L,.), and a sixth (described below) resulting from "timing out" (i.e., remaining at an internal state with pen up for a certain duration). Thus each state has six possible successors. If, in Figure 6, a particular direction is not noted as a link path from a state, it means that that direction returns the marker to that state. For example, state DD (Figure 6, lower-left) is entered via a D link path, but subsequent D signals will not move the marker, nor will a *U* signal. Only an *L*, *R*, or . signal following the D will advance the marker.

Let us now consider how this machine implements the specific character recognition sequence for the character 1. Although 1 is nominally described as a single downstroke followed by a pen lift, the logic can accommodate a wide variation of sequences that are equivalent. For example, starting from the INIT state, sequences (D,.), (U,D,.), (R,D,.), and (U,R,D,.) are all equivalent in moving the marker to state AA. Furthermore, the (R,D,.) and (U,R,D,.) sequences can be terminated with an R stroke — i.e., (R,D,R,.) and (U,R,D,R,.) — without affecting the final termination of the marker at Node AA. The (D,.) and (U,D,.) strokes can be terminated with a U as well as an R stroke — e.g., (D,U,.) or (U,D,U,.). Thus all these sequences are equivalent to the (D,.) sequence. The acceptable ways to make the basic downstroke of the character 1 are summarized in Figure 7a, which illustrates the ability of the logic to ignore the inevitable glitches produced by human writers.

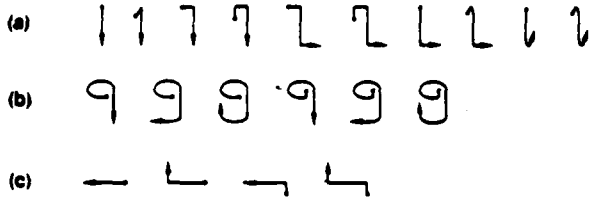


Figure 7. **The basic variations allowed in making the characters 1, 9, and erase.**

Provision for this range of spurious initial and final signals, however, produces a conflict with the 7 — basically an (R,D,.) sequence — which would be treated as a 1. To avoid this conflict, the seven is completed with a cross stroke in the European manner.

This illustration of the crossing of a character introduces the problem of character segmentation. How does the system know whether an (R,D,.) sequence, for example, is to be a 1, or whether it will subsequently be crossed, meaning a 7? The conflict is resolved with the conditional output logic implemented at state AA. To follow this conditional output scheme, note that any of these cross-stroke sequences — (R,.), (U,R,.), (R,U,.), (U,R,U,.), or (U,.) — will advance the marker from Node AA through subsequent states to the 7 output port. Any other sequence implies that a 1 was intended and that the subsequent strokes were the beginning strokes of a new character. (Note that the cross-stroke sequences are therefore not allowable as the beginning strokes of a character.) Thus, a left stroke should signal a 1 and move the marker back to state IL, where an initial left movement would have moved the marker from an INIT start. Of course, many other direction sequences can follow a 1. These all signal a 1 and move the marker to the appropriate internal state. Consider, for example, a 1 following a 1. The first 1 will energize state AA. The subsequent (D,.) sequence (or any of its equivalences) will energize output port (1,ID), thus signaling a 1 (the first 1) and return the marker directly to state AA via state ID. In other words, a sequence of 1's will continually cycle through the output port (1,ID) and then back to state AA through state ID.

Also shown in Figure 6 is a portion of the logic associated with the detection of digit 9 and the erase character. The former is described nominally as an (L,U,R,D,.) sequence, although variations are also permitted. In particular, the sequence can begin with a down stroke, i.e., (D, L,), and can end with a left stroke or even with a left, up stroke. Accepted variations are shown in Figure 7b. The erase character is basically a left stroke with the allowed variations shown in Figure 7c.

We have noted that, from any state, an arbitrary output code can be signaled and the marker advanced to any other state. Let us note one other special capability: timing out. A timing function is provided that measures the elapsed time since the last pen-up. If the elapsed time before the next pen-down is greater than some specified magnitude (e.g., 500 msec), the marker will automatically return to the INIT state, and an arbitrary output can be signaled. This is handled by treating the timing-outs as a sixth link path from each state. Without this special timing action a 1, for example, as the last character in a string would cause the marker simply to advance to and remain at state AA. With timing-out, a 1 output is automatically produced and the marker returned to the INIT state.

ROM implementation

The recognition logic can thus be thought of as a state machine with five direction inputs (U,R,D,L,.), a time-out input, and a set of output codes (e.g., ASCII code words). A particularly straightforward synthesis can be achieved with ROM logic. The use of programmable ROMs is especially useful during the iteration of link-path structures, because changes can easily be made in the ROM content rather than in the hardware.

Each state of the machine is assigned a block of addresses that contains all the link-path connections to subsequent states plus the timing-out and conditional output operations. Because it is possible to energize conditionally an output port as well as advance to another state, each location can contain either an output code or a new state address (indicated by the most significant bit of the word).

An efficient synthesis of the system can be achieved with 8K (1024-word x 8-bit) IC chips. The most significant bit (MSB) of each address is reserved as a flag to indicate whether the subsequent 7 bits are to be treated as a state address (MSB = 0) or as a 7-bit output code (MSB = 1). The remaining 7 bits allow up to 2^7, or 128, state addresses. Each state, in turn, has 8 link paths: the five directions (U,R,D,L,.) plus three others discussed below. Thus, each state occupies eight addresses, and each 1024-word, 8-bit ROM can therefore implement 128 states, exactly the number addressable by the 7 bits. The numerics-only machine from which Figure 6 is abstracted contains approximately 75 states.

As shown in Figure 8, the 7 most significant bits of the address of any particular location in a ROM are specified by a state-address register (SAR), which specifies a block of eight sequential addresses. A 3-bit link-path address register (LAR) determines which of the eight cells within the state block is selected. The ten bits together specify one of the 1024 words of the ROM, the output of which contains either the next state or an output code. The LAR is set to 0, 1, 2, 3, or 4 according to whether the current pen direction is pen-up, up, right, down, or left respectively. It is set to 5 if the pen has timed out. If the ROM word currently addressed contains all zeros, the LAR is set to 6 on the next clock pulse. This is used to implement a conditional output when the next state is not INIT, as described below.

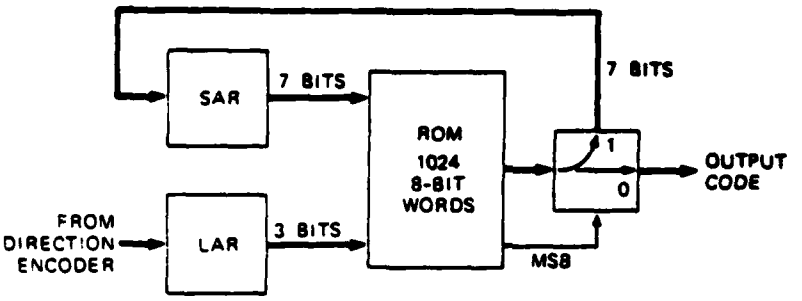


Figure 8.
ROM addressing scheme.

To follow the ROM synthesis in more detail, consider the hypothetical machine shown in Figure 9a. If we have reached state HH (i.e., SAR contains the address HH) and the pen turns downward, the very first clock cycle that recognizes the D direction will set the LAR to 3. At that indexed location is the address of state JJ, as shown in Figure 8b, which will be clocked into the state address register. The LAR, however, will not change as long as the pen continues to move downward. During that time, each clock cycle will address word 3 of state JJ, which contains the address of state JJ. That is, state JJ is entered via a D signal, and the marker will remain at JJ for as long as the D signal persists. (If the address at word 3 of state JJ were other than JJ, a sustained D direction would have caused the marker to move away from state JJ on the next clock cycle after entering state JJ.)

If the pen is subsequently moved to the right, the LAR will be set to 2, and the address for state KK will be fetched. If the pen is lifted, ".", the node address will remain unchanged (LAR index 0 also contains address JJ), but if the pen remains up for longer than the specified interval, the LAR will be set to 5, where the code for output character β is found, and the state address register will be reset to the address of the INIT state.

All time-outs and most normal outputs will produce a transition to the INIT state. The address of INIT is chosen to be SAR = 0, so that this state transition can be produced simply by clearing the SAR. Owing to this choice, the address of INIT does not have to be stored in the ROM.

For conditional outputs which do not return to INIT (called dual mode), it is necessary to store both the output code and the next state address. The implementation of this feature uses LAR index cells 6 and 7.

A dual-mode output is indicated when the contents of the selected ROM word are all zeros. For example, at state JJ the contents of LAR 1 and 4, i.e., U and L, are zero. If the pen moves in either of those directions, the all-zero ROM word will cause the LAR to be set to 6 for one cycle and then to 7 for the next cycle. During the first cycle, the code for character γ will be outputted (because the MSB of that word is 1); during the subsequent cycle, the address for state LL will be fetched. At state LL, a left movement will continue directly to state MM, and a U movement will continue directly through to state NN. That is, starting from state JJ, a U movement will move the marker to state LL and then, during the very next cycle, to state NN. Inserting the extra state, LL, avoids the more complicated conditional structure that would be necessary if we had to program the L transition from state JJ to one state and the U transition to a different state.

Other functions could be added to each state. For example, movements could be quantized into more than four directions, or different LAR locations could be addressed if the movement in a certain direction were greater or less than some specified duration. These added functions would, of course, require larger blocks of addresses for each state location.

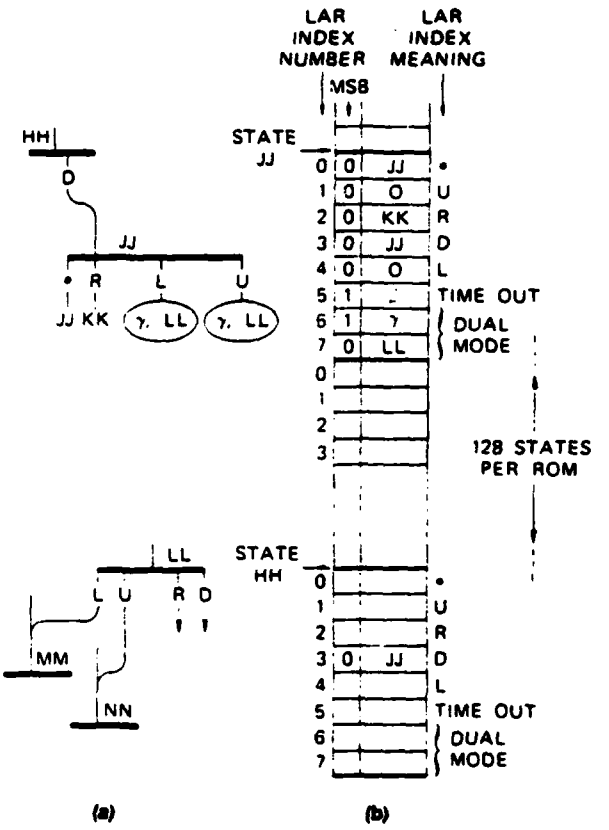


Figure 9. ROM realization of the state machine structure.

Although many trade-offs are possible — greater freedom can be allowed in one character at the expense of others — a state machine, whether realized in ROMs, PLA's, or in a microprocessor, is efficient in handling a wide range of variations without having to list or to account specifically for every allowed sequence, or even every element of each sequence. This is in contrast to a table look-up, which requires a complete listing of all allowed sequences. The design of recognition sequences in either case, however, is still largely ad hoc, and the partial structure illustrated in Figure 6 has evolved through many interactions to improve performance.

Performance characteristics

A major source of error for the novice is letting the pen point rest on the writing surface in a static position at the beginning or end of a character. A spurious pattern of direction signals is produced because there is strain at the pen tip even though there is no motion. To avoid these spurious signals, the user must learn to move the pen in the desired initial direction before (or just as) the pen touches the paper, and to continue moving in the final direction as the pen leaves the paper. Of course, many spurious patterns can be tolerated (logical "don't cares"), as has been shown in connection with Figure 6. Direction signals are ignored during an initial dead time beginning when the pressure threshold is first exceeded. This dead time (typically in the range of 50-100 msec) helps to ensure that the pen is moving in the desired initial direction before sampling actually begins.

By delaying the use of direction-change information, it is possible to ignore any final direction(s) that are shorter than a minimum duration. This form of back-end timing minimizes the effect of spurious tails at the end of strokes.

Another initial difficulty is learning to hold the pen vertically. Any tilt biases the force pattern in the direction of the tilt. With strong tilt to the left, for instance, the direction encoder could continue to signal "left," even though the pen were actually moving in another direction.

We have developed several effective aids to learning. Four direction lights continually signal the instantaneous direction of writing, as determined by the signal processor. Also, each character can be displayed on an accumulating alphanumeric visual display and/or repeated audibly by loudspeaker or earphone as it is recognized.

Most users quickly adapt to the smooth movements required as the pen touches and leaves the paper and to the need to hold the pen reasonably vertical, and error rates typically drop to a few percent within an hour or so of practice. After this learning period, surprisingly variable writing can be tolerated, as illustrated in Figure 10, which shows an array of characters written at one sitting by a single user, in which every character was correctly recognized. Note in particular that the system is inherently independent of character size and quite tolerant of sloppy printing.

Practical systems can be designed around this pen for relatively small character sets, e.g., the ten digits plus a few special characters such as erase and space. A state logic system for more than 40 alphanumeric characters has also been designed. Users experienced with the numerics-only set can perform reasonably well with this larger set. However, it is not yet clear whether a practical system with this many characters could be designed for a broad range of users.

Format control

Thus far we have considered only the problem of recognizing isolated characters as they are produced by the pen. However, as noted earlier, the system has no measure of absolute pen position in space. In using the pen to fill out a form, it is necessary to specify the box or area being filled out at any moment.

For use with common forms rather than free-format entry, the pen itself can enter format information. By letter code, the pen can specify to the system what fields of data are to be entered and in what order, how large each field is, and whether the field is numeric or alphanumeric. For final verification of data before entry to the computer system, the entered data can be displayed on a screen in the format of the form being filled out.

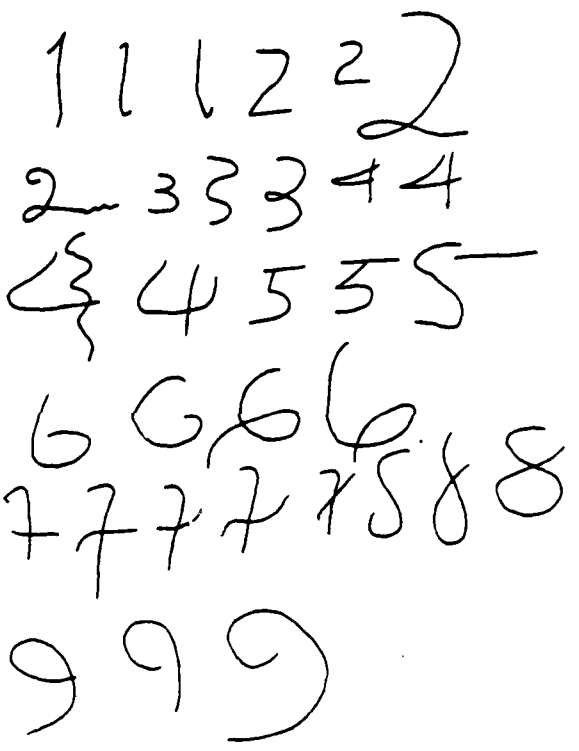


Figure 10. Random characters recognized without error. Note that the system is inherently independent of character size.

The pen, in other words, can be used in two modes within an integrated system. In the format entry mode, a computer processor is programmed by the pen to accept certain kinds of information in a particular format, as an intelligent terminal might be programmed. In the data entry mode, the system accepts the detected characters as data.

Discussion

The pen described in this paper permits a system design requiring no special writing surface or special writing environment.

In contrast to OCR schemes, which suffer from paper-related problems such as dirt smudges, breaks in the ink pattern, and folds in the paper, the scheme described here uses information derived from the pen itself, not from the writing on the paper. The final image is irrelevant to the character recognition process, and the paper can immediately be reduced to archival status. Because of the simplicity of the recognition logic and the elimination of special paper-handling requirements, the total system can be small and portable.

Static and dynamic methods of character recognition might be usefully complementary for very large character sets — e.g., Chinese script — that neither technique alone could handle. Characters having similar dynamic patterns but distinctly different static forms can be separated by static methods. For example, the letters P and D drawn as (D,..R,D,L,.) sequences are indistinguishable by the dynamic method discussed here, although statically they are easily distinguishable. Similarly, characters with

similar stroke configurations, i.e., similar static forms, in which the strokes are made in different sequences, can be distinguished by dynamic methods. In other words, certain dynamic information captured as the material is written may be useful during subsequent processing even if not adequate alone for real-time processing.

The system delivers ASCII code words as output and is compatible, therefore, with computer teletype ports. The strain-gauge transducers need to be sampled only about 50 times per second, and only changes in direction need be transmitted to the logic processor. Thus, only a small amount of preprocessor circuitry need be connected with each pen, and the direction information can be transmitted with low bandwidth to a central processor. ■

References

1. Talos Cybergraphic Tablet, Talos Systems, Inc., 7311 East Evans Road, Scottsdale, Arizona 85260
2. Graf/pen, Science Accessories Corporation, 65 Station Street, Southport, Connecticut 06490.
3. H. D. Crane, "Sequence detection using all-magnetic circuits," *IEEE Transactions on Electronic Computers*, Vol. EC-9, No. 2, June 1960, pp. 155-160.
4. Paul M. DeRusso, Robert J. Roy, and Charles M. Close, *State Variables for Engineers*, John Wiley & Sons, New York (1967).
5. Christopher R. Clare, *Designing Logic Systems Using State Machines*, McGraw-Hill, New York (1973).

Acknowledgments

The authors are indebted to Stanford Research Institute personnel Gerry Andeen and Jon Taenzer, who worked on the strain gauge array, and Carroll Steele, who developed the electronics. Ken Scott and Clint Hurd of Xebec Systems, Inc., helped in optimizing the state logic structure.

Hewitt D. Crane is a staff scientist in Stanford Research Institute's Information Science and Engineering Division. His experience includes early computer device and system design. Recently, he has been involved in the development of a pen system for data entry and dynamic signature verification and in novel instrumentation for vision research. He received a BSEE degree from Columbia University (1949) and a PhD in electrical engineering from Stanford University (1959).

Dr. Crane has more than 40 published papers and over 50 issued patents in a number of different fields. He received an Industrial Research IR-100 award in 1974 for the development of the data-entry pen system.

Robert E. Savoie is manager of the Research and Development Department of Telesensory Systems, Inc., Palo Alto, California, where he is responsible for the development of new sensory aids for the handicapped. His professional interests are in artificial intelligence, human factors, and new product decisions. Previous experience includes research in visual psychophysics, ophthalmic instrumentation, and pattern recognition at Stanford Research Institute, Menlo Park, California.

Dr. Savoie received his BA (magna cum laude) and his BSEE from Rice University in 1966 and 1967. He was a National Science Foundation Graduate Fellow at Stanford University, where he received his MSEE (1967) and PhD (1972) in electrical engineering.

Appendix C

THE SRI PEN SYSTEM FOR AUTOMATIC SIGNATURE VERIFICATION

THE SRI PEN SYSTEM FOR AUTOMATIC SIGNATURE VERIFICATION

Hewitt D. Crane, Daniel E. Wolf and John S. Ostrem
Stanford Research Institute, Menlo Park, California 94025

I. INTRODUCTION

A need has been growing in recent years for a practical, automatic, personal identification system in both government and private business. Applications range from government high security, such as controlling access to sensitive areas, to protection of access to computer facilities and data banks. Most of the methods of personal identification so far developed have been based on fingerprints, voice, personal identification numbers (PIN), physical features such as hand geometry, and naturally, the handwritten signature. Signature verification is one of the most promising techniques, considering psychological acceptance, technical feasibility, and cost.

By "signature verification" we mean the following: The person whose identity is to be verified gives a name or ID number and writes a signature, which will be referred to as the test signature. The test signature is then compared with a computer-stored representation, called the template, of the signature corresponding to the given name or ID number. If the test signature is "close" enough to the template by some appropriate measure, the person's identity is verified; if not, he is judged an imposter.

Automatic signature verification requires a representation of the written signature in a form suitable for computer input and subsequent data processing. There are basically two ways to obtain such a signature representation. One is to scan the signature optically after it has been written; this technique is similar in principle to that used for optical character recognition. However, optical scanning devices usually are bulky, expensive, and generally unsuited for real-time applications of signature verification. A more attractive and useful approach is to have either the writing device or the writing surface generate signals representative of the signature while it is being written.

In this paper, we describe an automatic, real-time signature verification system that has been developed at Stanford Research Institute (SRI). We present Type I (true-signer rejection) and Type II (forger acceptance) error rates as determined from tests on a first data base of true signatures and attempted forgeries. In the discussion in Section VI we state why we believe the results presented are conservative and will be improved in the future.

II. SRI THREE-AXIS PEN

The SRI signature verification system uses a strain-gauge-instrumented ballpoint pen, shown in Figure 1, that was developed by Crane at SRI.[1] A small array of strain-gauges near the ballpoint tip generates three electrical signals that are representative of the instantaneous three-dimensional drag force at the writing tip. Specifically, three independent orthogonal components of the total drag force are measured: downward force perpendicular to the plane of the writing surface (henceforth called pressure, or P), far/near force in the plane of the writing surface (called Y), and left/right force in the plane of the writing surface (called X). Each of the three force signals has a high signal-to-noise ratio. The pen has an ordinary writing tip and it requires no special writing surface.

III. SIGNATURE VERIFICATION: PARAMETERS METHOD

The signature verification process is based on a template matching procedure in which the P, X, and Y force signals generated during the writing of the test signature are compared against the P, X, and Y force signals of the appropriate template stored in a computer. The comparison can be made in many ways. But in general, a numerical measure of the "closeness" of the test signature to the template is computed and compared against a preset value, which we call the decision threshold. If the numerical measure of "closeness" is less than or equal to the decision threshold, then the test signature is judged to be a true signature. If the test signature is greater than the decision threshold, it is judged a forgery. A parameters (or features) technique computes as the numerical measure of closeness a normalized vector difference between a set of feature values extracted from the test signature and the corresponding feature values of the appropriate template. This technique is computationally efficient, requires only a small amount of template storage for each system user, and can be implemented in a stand-alone microprocessor unit. Other, more sophisticated verification techniques are discussed briefly in Section VI.

In the parameters technique, a number of parameter values (features) are extracted from the three continuous force-signals generated by the pen during the writing. These features include the total time of the signature, the time the pen is on the paper, the time the pen is off the paper, the average force in each of the three dimensions, the average energies, the average angle of writing, and many others. It is likely that not all of the extracted features will be equally effective for discriminating between true signatures and attempted forgeries. Also, it is desirable to reduce the number of features to save computation time and template storage space. For these reasons, a feature selection technique is used to select those features most effective (resulting in the least probability of error) in discriminating between true signatures and forgeries. Thus far in our analysis, we have examined more than fifty such features. By application of a standard F-ratio method of analysis (see reference 2), typically we reduce the number of features to between 10 and 20. Either a uniform set of features is used for all subjects or sets that are personalized for each subject are used.

Given a set of features, the decision-making algorithm used for deciding if a particular test signature is a true signature or a forgery is as follows: When a test signature is written, a value for each of the features is extracted from the P, X, and Y signals. The test signature may thus be represented by a feature vector $\vec{s}$, defined as

$$\vec{s} = \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_f \end{bmatrix} \qquad (1)$$

where f is the number of features extracted from the test signature and s_i is the value of the i^{th} feature. To determine if the test signature is a true signature or a forgery, the feature vector, $\vec{s}$, must be compared

with the appropriate template vector, $\vec{t}$. A template vector $\vec{t}$ must be obtained for each system user. This requirement necessitates a simple enrollment procedure in which each user signs several of his true signatures. The template vector for each user is constructed by averaging the N signatures obtained during the enrollment procedure. Thus, for a particular user

$$\vec{t} = \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ t_f \end{bmatrix} \qquad (2)$$

where $$t_i = \frac{1}{N} \sum_{j=1}^{N} t_{ij} \qquad (3)$$

is the average value of the i^{th} feature and t_{ij} is the value of the i^{th} feature for the j^{th} true signature obtained during an enrollment procedure in which N signatures are taken.

A template covariance matrix, C, can be computed as

$$C = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1f}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & & \cdot \\ \cdot & & \ddots & \cdot \\ \cdot & & & \cdot \\ \sigma_{f1}^2 & \cdots & & \sigma_{ff}^2 \end{bmatrix} \qquad (4)$$

where $$\sigma_{ik}^2 = \frac{1}{N-1} \sum_{j=1}^{N} (t_{ij} - t_i)(t_{kj} - t_k) \qquad (5)$$

are the unbiased estimators of the elements of the covariance matrix, C. The diagonal elements are the variance of the respective parameters; σ_{ii}^2 is the variance of the i^{th} feature, and $\sigma_{ii} = \sqrt{\sigma_{ii}^2}$ is the corresponding standard deviation.

Under the explicit assumption that the set of f features is distributed jointly as a multivariate Gaussian density, it can be shown that an optimum rule for classifying a test signature as true or as a forgergy is the following.

Compute the distance metric

$$d(\vec{s}) = \sqrt{\frac{1}{f}(\vec{s}-\vec{t})^{\dagger} C^{-1} (\vec{s}-\vec{t})} \qquad (6)$$

(where † indicates the transpose operation and C^{-1} is the inverse of C), and declare that the test signature is true if $d(\vec{s})$ is equal to or less than the decision threshold and that otherwise, it is a forgery.

Unfortunately, the distance metric of Equation 6 has several disadvantages for application in a practical system if f is large. The matrix inversion of C may be quite time-consuming; considerable space for template space is required (since an entire covariance matrix must be stored for each user); and, most important, a large number of true signatures (typically, several times f signatures) is required to obtain a statistically confident estimate of each user covariance matrix, thus leading to a much-extended enrollment procedure. For these reasons, we employ a simpler form of distance metric, obtained by assuming that the features are mutually statistically independent. In this case, all the off-diagonal elements of the covariance matrix are zero (i.e., $\sigma_{ij}^2 = 0$ for $i \neq j$), and Equation 6 reduces to

$$d(\vec{s}) = \sqrt{\frac{1}{f} \sum_{i=1}^{f} \left(\frac{s_i - t_i}{\sigma_i} \right)^2}, \qquad (7)$$

where for convenience we have compressed the notation by setting $\sigma_{ii} = \sigma_i$.

The distance metric of Equation 7 is simple, fast to compute and requires only 5 to 10 true signatures for the user to be enrolled. However, some loss of performance is expected if the set of features has significant linear correlation.

In fact, it is probable that the signature verification features are not jointly distributed as a multivariate gaussian density. In this case, neither of the previously shown distance metrics are known to be optimum, and it is not clear that the distance metric given by Equation 7 will yield worse performance than the more complex distance metric given by Equation 6, even if the features are significantly linearly correlated. We therefore use the distance metric of Equation 7 because, even though it probably is not optimum, it is still a reasonable classification algorithm that has yielded good performance in prior studies, and has all the advantages previously mentioned for application to a practical signature verification system.

Using the distance metric of Equation 7 requires that two numbers (an average value and a standard deviation) be stored for each feature of a subject's template. Basing the analysis on, say, 10 features therefore requires storing 20 numbers per subject (approximately 200 bits). By selecting a set of features for each subject, it may be possible to use only 5 to 10 features per subject, both reducing storage requirements and improving performance.

IV. DATA BASE

A data base of true signatures and attempted forgeries has been obtained for the purpose of estimating the Type I and Type II error rates for this system. Sixteen persons selected randomly from a larger group of volunteers were subjects for the data base. Included were secretaries, research assistants and engineers. Each subject was given a set of written instructions describing the procedure for the sign-in sessions and was scheduled to appear for between one and three sign-in sessions per week over a period of three months for a total of 16 sign-in sessions. At each session, the subject signed his or her own signature three times and attempted two forgeries of one of the other data base members. For the forgery attempts, each subject was given several copies of the signatures of all the other subjects, a written form that stated that the signature verification system based the true/forgery decision on matching the forces and motions involved in writing a signature, and was encouraged to practice prior to the formal forgery attempts. The subjects were not given any feedback either on whether their true signatures were verified or whether their forgery attempts were accepted or rejected.

The P, X, and Y force signals for each of the true signatures and attempted forgeries were stored on magnetic tape. At the conclusion of the specified time period, about 800 true signatures and 425 attempted forgeries had been collected.

V. RESULTS

The data were analyzed in several ways.

The 800 true signatures were divided randomly into two groups of 400. The selection of features for each subject was made using an F-ratio analysis to choose those features that were most effective in discriminating his or her true signatures against all other true signatures of the first group of 400. (This technique of feature selection could be automated in a practical signature verification system.) The actual error rate calculations then were performed using the second group of 400 true signatures together with all of the 425 forgeries. A template for each subject was constructed by averaging together all his or her true signatures in the second group of true data. The calculated error rates, using the decision-making algorithm of Equation 7, are shown in Figures 2, 3, and 4.

To calculate the Type I error rate, a value of the distance metric must be computed for each true signature. However, if the distance metric for a particular true signature is computed using a template that includes that particular signature, the resultant error rate will be overly optimistic. For this reason, we subtracted each true signature from the template when its distance metric was computed.

Toward the end of the data base, we observed that the error rates seemed to increase. We believe this was because some subjects eventually lost interest and became careless, owing to the lack of feedback and motivation in the experimental design. This perhaps can be minimized by better experimental design and may well not be an important factor in an operational system. For this reason, we considered it of interest to perform again the error rate analysis, but this time excluding some of the later data-taking sessions. Thus, error-rate calculations that use only the first 40 data files, out of a total of 58, are presented in Figure 5.

Finally, we analyzed the first 40 data files using pressure- and timing-related features only, to test how much improvement can be expected from a system that utilizes a 3-axis pen over a single-axis (i.e., pressure-only) pen. These results are shown in Figure 6.

All Data Files

Figures 2(a) and 2(b) are computer printouts in increments of 0.1 in RMS difference (we call the calculated value of the distance metric, Equation 7, the RMS difference) for the true signatures and forgeries of each subject. Figure 2(a) summarizes the true-signature data and Figure 2(b) displays the forgery data. From Figure 2(a), we see that the first subject (JAB) entered 28 true signatures (the sum of column 1), ranging in value of RMS difference from 0.8 to 1.7. The second subject (JLC) entered 25 true signatures (the sum of column 2), ranging from 0.9 to 1.6. From Figure 2(b), we see that there were 30 attempted forgeries of subject JAB, the closest having a value of 2.0. There also were 30 attempted forgeries of subject JLC, the closest having a value of 4.4.

Figure 3 is a distribution plot of the true and forgery values across all subjects. We see, for example, that 14 of the 393 true signatures had RMS difference values greater than 1.6, and 8 of the 425 forgeries had RMS difference values less than 1.8. Alternatively, 379 (or 96.4 percent) of the true signatures had RMS values less than 1.6, and 417 (or 98.1 percent) of the forgeries had RMS values greater than 1.8.

The overlap between the true and forgery data is the source of the Type I and Type II errors. The magnitude of each type of error is a function of the value of the threshold for the RMS difference that is chosen, above which a signature is called false and below which it is called true. Magnitude of error as a function of threshold is shown in Figure 4 for the data of Figure 3. Note that the Type I and Type II errors have an equal value (1.7 percent) at an RMS threshold level of 1.75.

First 40 Data Files

Figure 5 shows the error results when only the first 40 data files are considered. Actually, the equal Type I/Type II error rate is the same (1.7 percent), although the Type II error rate falls more rapidly with lower threshold values. Thus, at a threshold RMS level of 1.8, there is a 0.7 percent Type II error rate (i.e., forgery acceptance) and 2.3 percent Type I error rate (i.e., true signature rejection).

Pressure and Timing Parameters Only

Figure 6 shows the error rate plots when the 40 data files of the previous section are rerun with all of the X-related and Y-related parameters deleted. The equal error rate is 5 percent.

VI. DISCUSSION

These results must be treated as strictly preliminary.

No one involved with the development of the pen or signature verification system was also involved in any of the data-taking sessions. In this way, we hoped to eliminate any biasing of results that might have been caused, for example, by subconscious coaching of the subjects by those who knew the system best. However, we would do some things differently in developing another data base. First, we would choose a different location for the test (a number of subjects complained after the data base was completed that the computer room in which the data was taken was very cold and their hands felt stiff). Second, we would shorten the time over which the data is collected or try to increase the motivation of the subjects. We found that the true-signer templates tended to develop larger standard deviations toward the end of the data base collection period, probably because of the lack of motivation and resultant loss of interest noted earlier. Because of the greater template variances, the forgery-acceptance rate increased. For this reason, we expected significantly better results from the first portion of the data base, although the curves of Figure 5 do not show as much improvement as might have been predicted.

Conservative Aspects of the Results

We believe that these results are conservative in three major ways. First, the data was taken with an early, semi-production model of pen, which, unfortunately, had a round body. Subsequently, we have obtained improved results with a triangular-shaped pen of the type shown in Figure 1. The X and Y signals from the pen are sensitive to the angle of "roll" about the main axis of the pen. With a triangular body, the subject grasps the pen much more consistently every time. Of course, any parameters that are sensitive to roll could be eliminated from the analysis, but including them

results in a great improvement in performance (although consistent roll angle should not have much effect on pressure and timing parameters). This result was first noticed in a formal way when taking a small data base of Chinese signatures using native-born Chinese. Chinese characteristically hold a pen loosely, somewhat like an artist's paint brush. Performance with a round-bodied pen--initially poor--became comparable to the results shown here when a triangular-shaped pen was substituted while the group of subjects remained the same.

Second, for the purpose of analyzing this data base, an individual's template was made by averaging true signatures taken over a period of three months. A more likely procedure, at least in some operating systems, would be to use the first half-dozen or so signatures for each subject as his or her template, and update periodically by averaging in signatures that are verified with an RMS difference value less than, say, 1.0 or 1.2. In this way, the template automatically would track any slow changes in the subject's signature, and the subject's standard deviation values would tend to be smaller, making his signature more difficult to forge. However, we were not able to reanalyze the data on that basis for this paper.

Finally, while the F-ratio technique for feature selection is simple and efficient, it is by no means optimal. Its primary disadvantage is that it evaluates separately the discrimination power of each feature, ignoring the effects of interfeature correlations. Also, the F-ratio has no definite relation to the probability of error, except when the distribution of values for a feature for the true signature and forgeries are both Gaussian and distributed with equal variances. Therefore, we believe that the process of feature selection can be improved, and that the error rates probably can be reduced. We have begun to explore other methods for selecting sets of features, which will be reported at a later time.

Other Forms of "Signature"

Our discussion has emphasized actual signatures, but the system works well also with symbols of any form, such as the user's initials, the digits from 0-9, or one's telephone number. No formal data, however, has thus far been taken with other than normal signatures.

Other Forms of Devices

We have described a three-axis pen as an input device to a signature verification system. A three-axis platen system also has been developed at SRI for this purpose.[3] With this device, the user can write with an ordinary pen or pencil. Such a system might have significant advantages in certain applications, although informal data show that it may yield somewhat poorer performance than a system that utilizes a pen. In fact, its performance is likely to stand intermediate between a one-axis (pressure-only) pen and a three-axis pen. A one-axis pen is completely insensitive to X and Y forces. While a three-axis platen does generate X and Y signals, the signals are independent of the way the pen is held by the user. For instance, a line drawn from left to right on the platen will generate a pure X signal, regardless of pen orientation. With the three-axis pen, however, the coordinate system is attached to the pen, and therefore the X and Y signals are dependent on pen orientation. For instance, left-handed and right-handed users typically have a 180-degree shift in X,Y orientation. In other words, a three-axis pen provides more information with which to distinguish writers. In fact, the choice of all right-handed subjects in the data base is yet another conservative aspect of the results, inasmuch as lefthanded and righthanded users generally are easily distinguished.

Correlation Methods

We have developed also a correlation method for signature verification. In this method, the P, X, and Y time series force-signals of a test signature are correlated mathematically against the appropriate P, X, and Y template signals. If the test signature's correlation is greater than some preassigned threshold, it is judged true, and, if not, it is judged a forgery. However, straight mathematical correlation often yields poor results because of the normal variations in different true signatures. Even though the test signature and template P, X, and Y signals may be highly correlated by a subjective, visual comparison, small time shifts within the test signature P, X, and Y signals can cause important phase shifts with respect to the template P, X, and Y signals. To compensate for this effect, we have developed a number of techniques based on what we call "rubbery" correlation. In these methods, an automatic two-dimensional fitting procedure is used to find an optimal match between the template and test signatures, allowing time base translation and time warping (stretch and contraction) of the test signature P, X, and Y signals. These procedures can be applied independently to different parts of the signature--for instance, applied to the first half of the template and test signals and then independently to the second half of the signals; or the analysis can be done in thirds.

This method requires approximately ten times as much storage per subject (several thousand rather than several hundred bits per user) but has the potential to yield significantly better performance than the parameters method. With correlation, even if a potential forger has all of the raw signal data available, he would have to be able to translate the 3-axis visual information into appropriate muscle responses with great accuracy. Preliminary results show that it is very difficult for even a determined individual to learn to make such a match. These rubbery correlation methods will be reported in a future paper.

We believe that for some applications, and depending on the degree of performance required, there may be value in using both the quick-and-easy parameters method and the more sophisticated correlation methods. For instance, the methods of analysis have a degree of independence so that their simultaneous application should result in improved true/forgery discrimination.

VII. SUMMARY

We have described an automatic signature verification system. The system uses a ballpoint pen equipped with an array of strain gauges mounted near the ballpoint tip. The gauges record the instantaneous three-dimensional, or three-axis, drag force generated at the tip during writing, and these signals are utilized by the verification system. In other words, the system analyzes the dynamics of writing rather than the static image produced by the pen. In fact, the system makes attempting to trace someone else's signature one of the worst possible strategies for forgery.

We have described a method of analysis called "parameters method." It is computationally simple and can be realized with current microprocessor technology. Templates for each user consist of approximately 200 bits which can be stored in a central data file or be encoded on a card carried by the user.

In this first reported data base, we have found equal Type I/Type II error rates in the range of 1 to 2 percent. We have stated why we believe these results are conservative. First, a round-bodied pen was used in collecting the data; we find much better performance with a pen that has a triangular-shaped body, which tends to be held more consistently each time. Second, the true-signature templates for the error analysis were formed from true signatures taken over the entire collection period of several months; more practical template-making procedures likely would utilize only the subject's most recent signatures, which generally lead to much "tighter" templates (i.e., smaller standard deviation values, which are more difficult to forge). Third, a straightforward F-ratio analysis technique was used for selecting features. However, this is not an optimal method. Currently, we are exploring methods that we hope will lead to an automatic means of selecting optimum feature sets that likely will be different for each user.

We have noted also a method of correlation analysis. This method requires about ten times as much template storage per user, but is a more effective method of analysis. Both methods may be applied simultaneously.

VIII. ACKNOWLEDGMENTS

This research was supported by SRI research and development funds.

We would like to thank Kathryn Mouton, who organized and collected the signature data, and the 16 volunteers who participated as data base subjects.

IX. REFERENCES

1. H.D. Crane and R.E. Savoie, "An on-line data entry system for hand-printed characters," in Computer, pp. 43-50, March 1977.

2. The F-ratio technique for feature selection is well-established and is described in many textbooks. For example, see W.J. Dixon and F.J. Massey, Introduction to Statistical Analysis, third edition, McGraw-Hill, 1969, Chapter 10; G.W. Snedecor and W.G. Cochran, Statistical Methods, sixth edition, the Iowa State University Press, 1967, Chapter 14; and D.E. Bailey, Probability and Statistics, Wiley & Sons, Inc., 1971, Chapters 17 to 19.

3. U.S. Patent 3,988,934 issued November 2, 1976.

Figure 1 SRI 3-axis pen

PROGRAM DISCRIM, VERSION 3, 18 MAR 77 JOBID: AEXAASQ MAR 29, 1977

FREQUENCY OF ATTAINING RMS VALUES: TRUE DATA

RMS DIFF	JAB	JLC	KMC	LED	REF	SEG	JNH	EMK	PGK	FJM	BPM	JRS	GLS	JW	ECW	Total
0.0 to .1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.1 to .2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.2 to .3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.3 to .4	0	0	1	1	0	0	6	1	0	1	1	0	0	0	0	11
.4 to .5	0	0	1	1	1	0	6	0	0	2	5	0	1	0	1	18
.5 to .6	0	1	3	3	2	2	3	3	1	0	2	0	5	1	1	27
.6 to .7	6	2	6	1	5	3	4	1	1	2	6	0	3	4	1	45
.7 to .8	1	6	1	3	2	1	2	5	4	6	5	2	2	6	6	52
.8 to .9	5	5	3	4	5	2	2	3	6	5	2	5	6	3	3	59
.9 to 1.0	4	1	1	3	2	2	1	1	5	4	2	7	1	2	3	39
1.0 to 1.1	3	3	1	1	1	3	2	2	1	3	1	6	3	5	4	39
1.1 to 1.2	1	1	4	4	0	1	1	4	3	2	0	5	1	0	2	29
1.2 to 1.3	2	1	0	1	2	2	1	1	0	0	0	1	1	1	3	16
1.3 to 1.4	3	1	1	3	1	0	0	3	2	0	1	3	2	0	1	21
1.4 to 1.5	0	2	0	1	0	1	0	2	1	1	1	0	1	0	0	10
1.5 to 1.6	1	2	1	2	1	1	0	0	0	1	1	0	1	2	0	13
1.6 to 1.7	2	0	1	0	0	0	0	0	0	1	0	0	0	0	1	5
1.7 to 1.8	0	0	1	0	0	1	0	0	0	0	0	0	0	1	2	5
1.8 to 1.9	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
1.9 to 2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0 to 2.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
2.1 to 2.2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
2.2 to 2.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.3 to 2.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.4 to 2.5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
2.5 to 2.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.6 to 2.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.7 to 2.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.8 to 2.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.9 to 3.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0 to 3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.1 to 3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.2 to 3.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.3 to 3.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4 to 3.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.5 to 3.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.6 to 3.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.7 to 3.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.8 to 3.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.9 to 4.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0 to 4.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.1 to 4.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.2 to 4.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.3 to 4.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.4 to 4.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.5 to 4.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.6 to 4.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.7 to 4.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.8 to 4.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.9 to 5.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	28	28	25	28	22	20	28	27	25	28	27	29	28	25	28	393

Figure 2(a) Computer print-outs for true data.

PROGRAM DISCRIM, VERSION 3, 18 MAR 77 JOBID: AWXAASQ MAR 29, 1977

FREQUENCY OF ATTAINING RMS VALUES: FORGERY DATA

RMS DIFF	JAB	JLC	KMC	LED	REF	SEG	JNH	EMK	PGK	FJM	BPM	JRS	GLS	JT	ECW	TOTAL
0.0 to .1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.1 to .2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.2 to .3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.3 to .4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.4 to .5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.5 to .6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.6 to .7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.7 to .8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.8 to .9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
.9 to 1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0 to 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.1 to 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.2 to 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.3 to 1.4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1.4 to 1.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1.5 to 1.6	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	3
1.6 to 1.7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
1.7 to 1.8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1.8 to 1.9	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	2
1.9 to 2.0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	4
2.0 to 2.1	1	0	0	0	0	2	0	1	0	1	0	0	1	1	0	7
2.1 to 2.2	0	0	0	0	0	1	0	0	1	1	0	0	0	3	0	6
2.2 to 2.3	0	0	0	0	0	1	0	0	0	1	4	0	3	0	1	10
2.3 to 2.4	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	5
2.4 to 2.5	0	0	0	0	0	1	0	0	0	1	1	0	0	2	1	6
2.5 to 2.6	0	0	1	1	0	1	1	2	0	1	1	0	1	2	0	11
2.6 to 2.7	0	0	1	1	2	4	1	2	0	0	0	0	0	1	0	12
2.7 to 2.8	0	0	1	0	0	3	1	1	0	1	1	0	0	0	3	11
2.8 to 2.9	0	0	0	2	0	0	0	1	0	0	1	0	1	1	0	6
2.9 to 3.0	2	0	1	1	2	2	0	0	0	0	2	2	0	0	0	12
3.0 to 3.1	2	0	0	1	1	0	1	0	0	0	1	1	1	2	0	10
3.1 to 3.2	1	0	0	1	3	1	1	1	0	1	0	0	0	0	0	9
3.2 to 3.3	0	0	1	0	1	0	0	1	0	1	0	1	1	0	1	7
3.3 to 3.4	0	0	1	0	0	0	1	0	0	1	3	1	0	1	1	9
3.4 to 3.5	2	0	3	0	1	0	1	1	0	1	1	1	0	0	0	11
3.5 to 3.6	1	0	3	2	2	0	1	1	0	1	0	0	2	0	0	13
3.6 to 3.7	2	0	0	1	1	1	0	1	0	1	0	1	0	0	0	8
3.7 to 3.8	1	0	2	0	0	1	0	2	1	0	0	1	0	2	0	10
3.8 to 3.9	1	0	0	0	0	0	0	1	0	1	0	1	0	0	1	5
3.9 to 4.0	0	0	0	0	3	0	1	1	1	1	0	0	0	0	1	8
4.0 to 4.1	1	0	0	0	1	1	0	0	1	1	0	1	1	0	0	7
4.1 to 4.2	0	0	0	2	2	0	1	0	0	4	0	1	2	0	0	12
4.2 to 4.3	0	0	0	1	2	0	0	1	0	0	0	0	0	3	0	7
4.3 to 4.4	1	0	0	0	0	0	0	2	0	0	0	1	0	1	0	5
4.4 to 4.5	0	2	0	1	0	1	0	1	0	0	2	1	1	1	0	10
4.5 to 4.6	1	0	1	2	0	1	1	0	2	1	3	0	0	1	0	13
4.6 to 4.7	0	0	0	0	1	0	1	1	0	1	1	0	0	1	1	7
4.7 to 4.8	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	4
4.8 to 4.9	2	0	0	0	1	0	0	2	2	1	0	1	0	0	0	9
4.9 to 5.0	11	28	15	14	7	2	18	6	22	2	3	12	2	3	16	161
Total	30	30	31	30	30	27	30	30	30	30	29	26	18	28	26	425

Figure 2(b) Computer print-outs for forgery data.

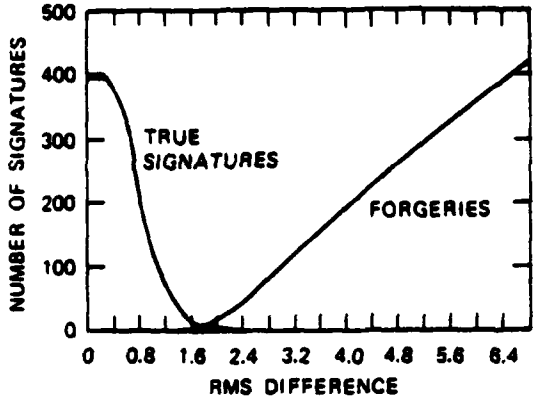


Figure 3 Accumulated true and forgery data as a function of RMS difference.

Figure 4 Type I and Type II errors as a function of the RMS difference threshold.

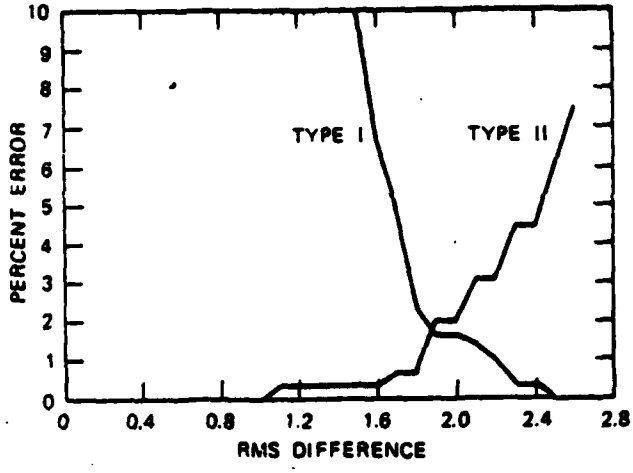
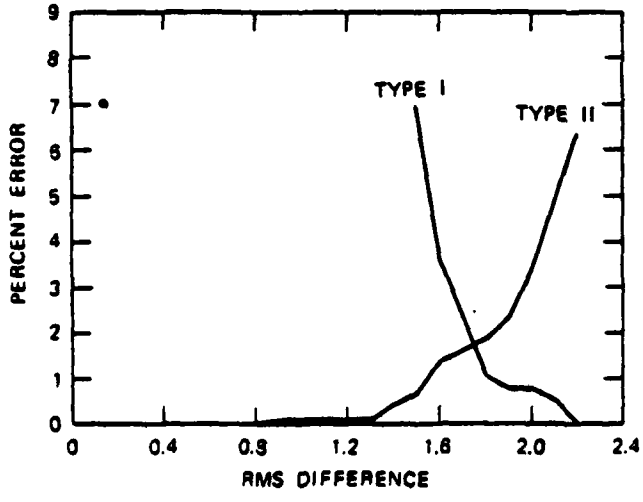
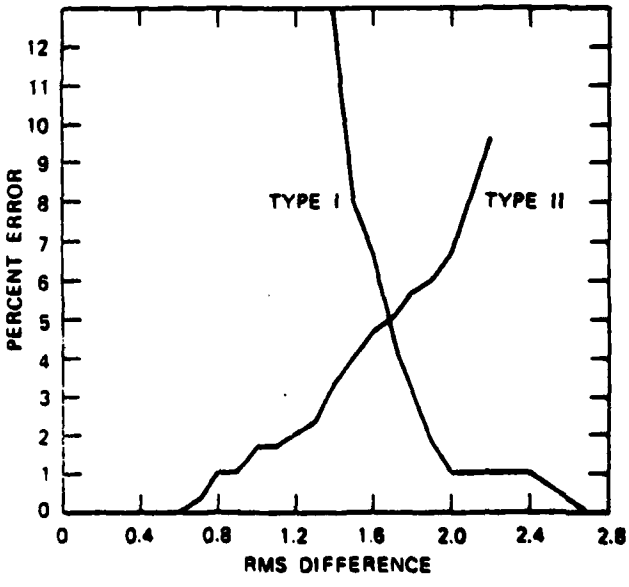


Figure 5 Type I and Type II errors for the first 40 data files (out of 58) as a function of the RMS difference threshold.

Figure 6 Type I and Type II errors for the first 40 data files using pressure- and timing-related parameters only.





Appendix D

MAXIMUM LIKELIHOOD ESTIMATION

MAXIMUM LIKELIHOOD ESTIMATION

In the main text of this report, a binomial distribution was used to describe the probability of R false rejects in T trials. For a single trial the equivalent relation is (for the ith trial)

$$\text{Prob}\left\{z_i\right\} = p^{z_i}(1-P)^{1-z_i} \quad ,$$

where P is the true population error rate to be estimated. In the above relation $z_i = 1$ if the ith trial is a false rejection, and $z_i = 0$ if the ith trial is a correct verification. Thus, for example, the probability that the ith trial is a false rejection is Prob $\left\{z_i = 1\right\} = P$, and the probability that it is a correct verification is Prob $\left\{z_i = 0\right\} = 1 - P$.

The goal is to derive an estimate for P that can be calculated using a verification data base and that in some sense best agrees with the actually observed data. The maximum likelihood approach yields one such estimate.* The first step in the procedure is to form the likelihood function L(P). Assuming independent trials, the joint probability distribution for T trials is

$$\text{Prob}\left\{z_1, z_2, \ldots, z_T\right\} = \prod_{k=1}^{T} P^{z_k}(1-P)^{1-z_k} \quad .$$

The likelihood function is defined as the logarithm of Prob $\left\{z_1, z_2, \ldots, z_T\right\}$:

$$L(P) = \log[\text{Prob}\left\{z_1, z_2, \ldots, z_T\right\}]$$

$$= \sum_{k=1}^{T} \log\left[P^{z_k}(1-P)^{1-z_k}\right]$$

$$= \left(\sum_{k=1}^{T} z_k\right) \log P + \left[\sum_{k=1}^{T} (1-z_k)\right] \log(1-P) \quad .$$

*The theoretical foundation of maximum likelihood estimation is too involved to treat here. For more details, see, for example, H. Cramer, Mathematical Methods of Statistics (Princeton University Press, 1951).

L(P) is maximized in the usual way by setting its derivative with respect to P equal to 0. This yields

$$\frac{1}{P}\sum_{k=1}^{T} z_k - \frac{1}{1-P}\left(T - \sum_{k=1}^{T} z_k\right) = 0 \quad ,$$

which implies that

$$P = \sum_{k=1}^{T} z_k/T \quad .$$

From the definition of z_k, we know that

$$\sum_{k=1}^{T} z_k$$

is simply the total number R of false rejects in T trials, so the maximum likelihood estimate of the error rate is

$$\hat{P} = \frac{R}{T} \quad .$$

Appendix E

CONFIDENCE LIMITS

CONFIDENCE LIMITS

In the main text of this report, the probability of R false rejections in T trials was expressed by the binomial distribution

$$\text{Prob}\left\{R\right\} = C_R^T \, P^R \, (1-P)^{T-R}$$

where P is the true error rate. An extimate of P that can be calculated from a data base of verification trials is

$$\hat{P} = \text{maximum likelihood estimate of } P = \frac{R}{T} \quad .$$

What is our confidence that the estimate $\hat{P}$ is a good approximation to the true error rate P? For simplicity, we assume that T is large* so that the binomial distribution can be approximated by a Gaussian distribution of variance $P(1-P)/T$. It can be shown (Snedecor and Cochran, 1967) that the probability that P lies between

$$\hat{P} - 1.96\sqrt{\hat{P}(1-\hat{P})/T} \text{ and } \hat{P} + 1.96\sqrt{\hat{P}(1-\hat{P})/T}$$

is approximately 95 percent. In other words, if we calculate $\hat{P}$ for a particular data base, we can be 95 percent certain that the true error P rate lies between the above limits. The two limits above are sometimes called the 95 percent confidence limits. The 99 percent confidence limits can be calculated simply by substituting 2.576 for 1.96.

Example--Suppose that 200 false rejects occur in 1,000 trials:

$$\hat{P} = \frac{200}{1000} = 0.2 \quad .$$

---

*For small T see Figure E-1.

Snedecor, G. W., and W. G. Cochran, Statistical Methods (Iowa State University Press, 1967), pp. 210-211.

We can be 99 percent certain that the true error rate is in the range

$$0.2 \pm 2.576 \sqrt{(.2)(.8)/1000} = 0.2 \pm 0.033 \quad .$$

That is, we are 99 percent certain that 16.7 *percent* $\leq P \leq$ 23.3 percent

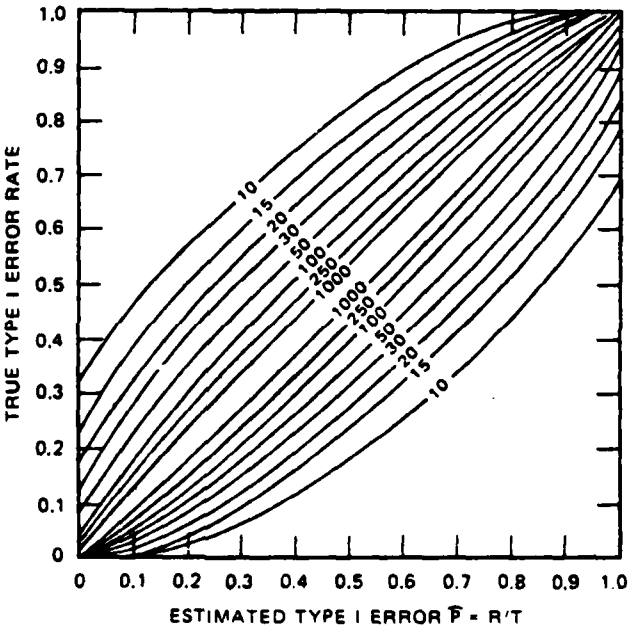
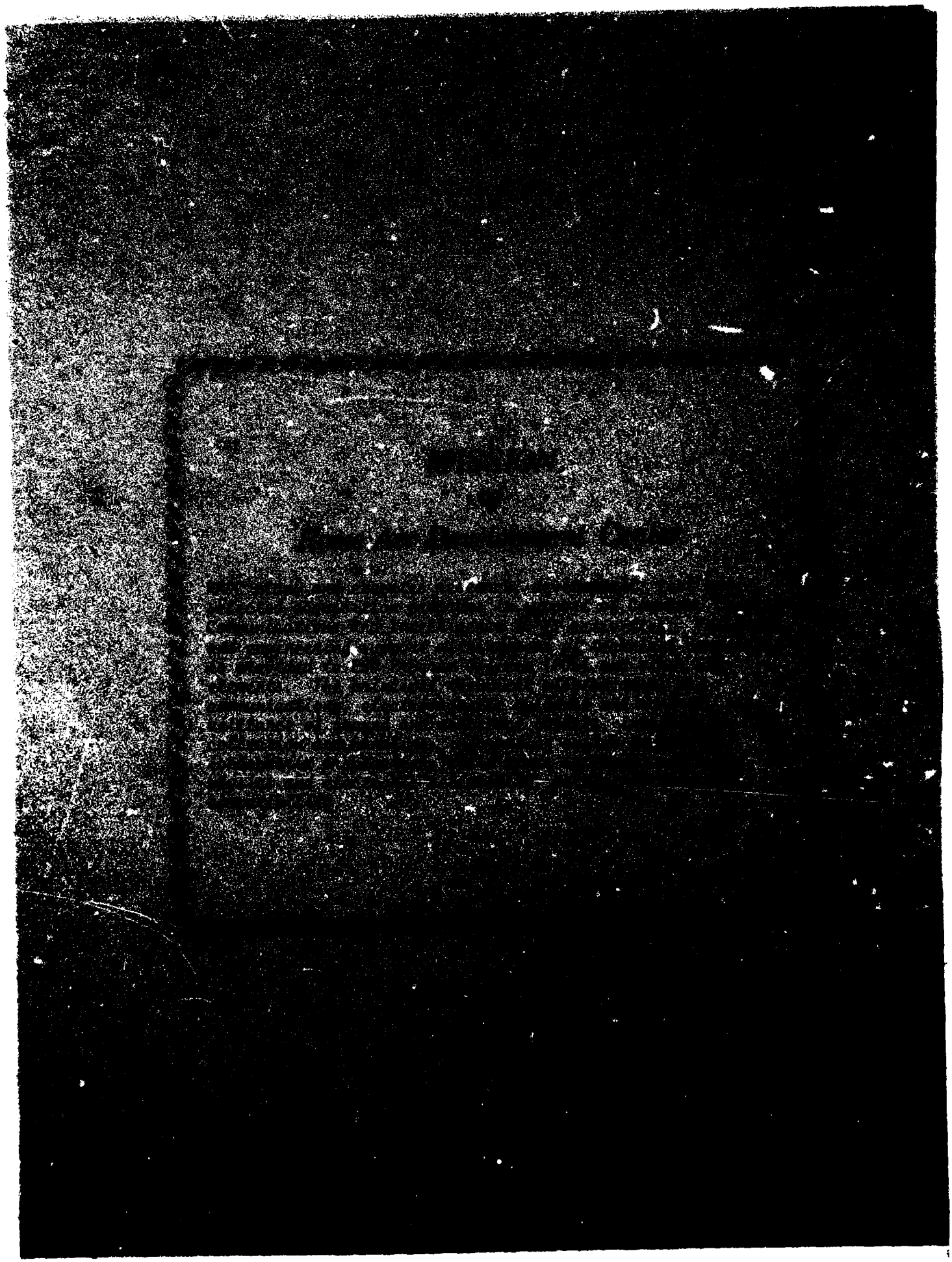


FIGURE E-1 CONFIDENCE LIMITS




DATE
ILME